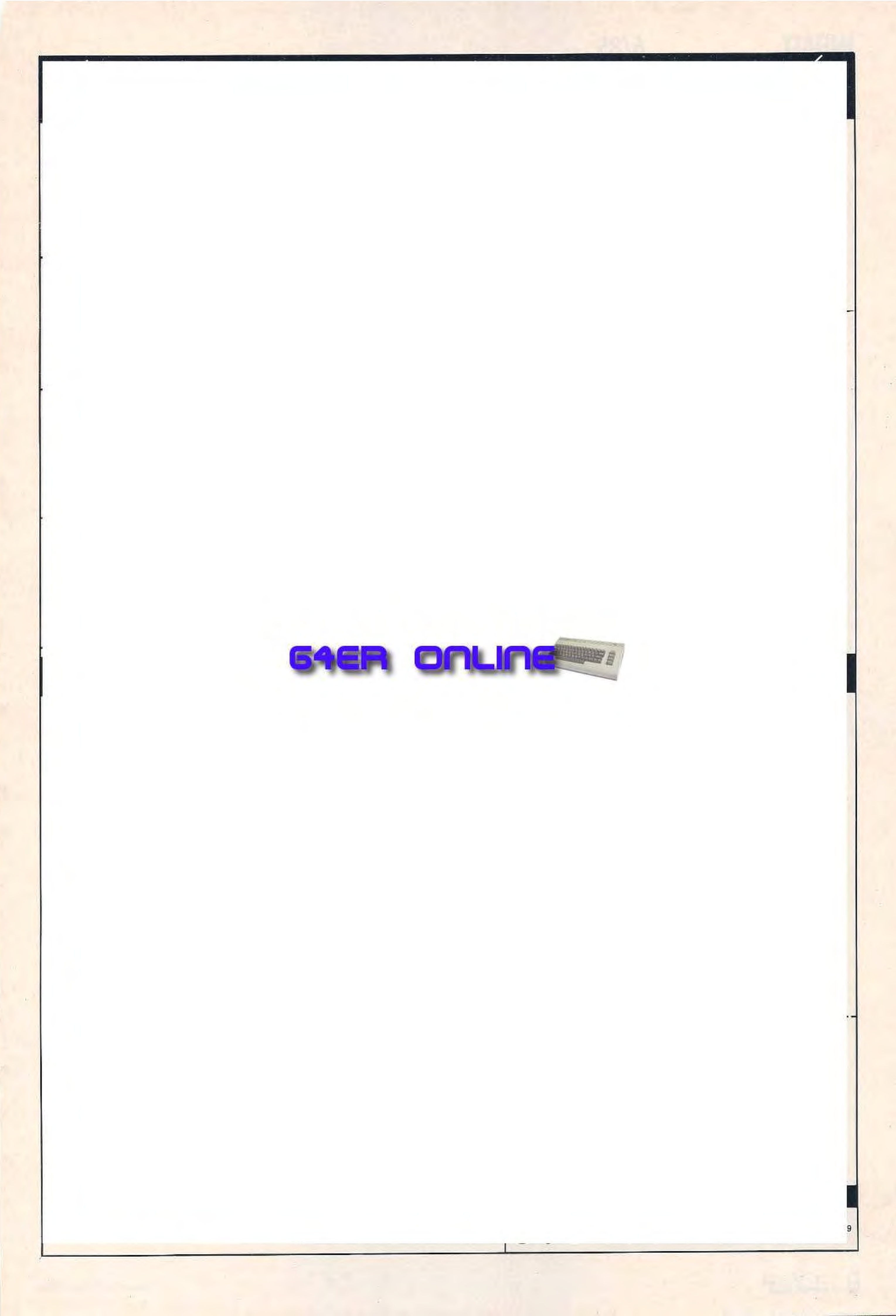


64'er
JUNI 1985
ÖS 55,—/sfr. 6,50 Lit 5.500/hfl 8,—/dkr 30,—
DM 6,50
6|85
DAS MAGAZIN FÜR COMPUTER-FANS
Eine Markt & Technik Publikation
Prototyp auf dem Prüfstand
PC 128
Erster ausführlicher Testbericht
Programme aus dem Telefon
★ DFÜ für Anfänger ★ Die Netze der Post ★ Terminalprogramme
Wettbewerb! Über 25000 Mark zu gewinnen
Typenraddrucker
Grundlagen, Test, Auswahlhilfen
Textverarbeitung
Textomat Plus und Vizawrite im Vergleich
COMMODORE
PC 128
Tips und Tricks für C 64, VC 20 und C 16
Macro-Basic – die etwas andere Basic-Erweiterung
★ Noch mehr Super-POKEs ★ Selbstbau:
Ungewöhnlicher Reset-Taster


64ER ONLINE


64ER ONLINE

Seite 152




**Wettbewerbe — über 25000 Mark zu gewinnen**

**In zwei Wettbewerben winken tolle Preise wie etwa ein PC 10, zwei Btx-Fernseher, eine Reise nach Berlin, zehn Floppy-Laufwerke, zwei Drucker MPS 802 und jede Menge Buchgutscheine. Nur wer mitmacht, kann gewinnen** **Seite 152**

Seite 32



**Typenraddrucker**

**Schön und langsam sind die Eigenschaften, die einen Typenraddrucker auszeichnen. Wollen Sie mit Ihrer Korrespondenz Eindruck machen, werden Sie um einen solchen Drucker nicht herumkommen. Typenradschreibmaschinen können einen Drucker ersetzen, wenn sie einen entsprechenden Anschluß haben.** **Seite 32**

Seite 16



**Test PC 128**

**Als erste Zeitschrift hatten wir die Möglichkeit, den neuen PC 128 über eine Woche lang direkt in der Redaktion auf Herz und Nieren zu testen. Unser ausführlicher Bericht zeigt, was im großen Bruder des C 64 alles drinsteckt.** **Seite 16**

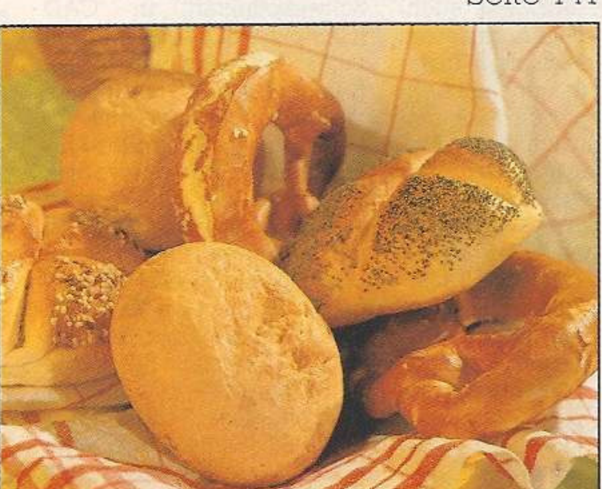
Seite 40



**Die Netze der Post**

**Datex-P, Btx, Telebox heißen die Datenübertragungsnetze der Bundespost. Was sich hinter diesen Begriffen verbirgt, wie man sich an diesen Netzen beteiligt und was sie kosten erfahren Sie ab Seite 40**

Seite 147



**Semmelservice mit dem C 64**

**Frühstück ans Bett. Das ist seit jeher eine feine Sache. Nicht ans Bett aber bis vor die Haustür werden in München die Semmeln gebracht. Die Tourenplanung, das Schreiben von Rechnungen und die Buchhaltung erledigt ein C 64. Seite 147**

Seite 140



**Directory-Manipulationen**

**Man muß nicht gerade mit dem Schweißbrenner an seine Disketten gehen, um sie vor fremden Zugriffen zu schützen. Wie man einfacher ein Directory manpuliert oder gar verschwinden läßt, erfahren Sie auf Seite 140**



## Der C 64 lebt weiter

Als Einsteigermodell, so sagt man bei Commodore, habe er noch zwei bis drei Jahre seine Berechtigung. Dann könnte — wegen des branchenüblichen Preisverfalls — der 128 die Rolle des Billig-Modells übernehmen. Dafür spricht eine einfache Überlegung: Es ist wenig wahrscheinlich, daß der Absatz in diesem Jahr schlagartig zusammenbricht, nachdem 1984 allein in Deutschland knapp eine halbe Million 64er verkauft wurden und es im Frühjahr schon wieder Lieferprobleme gab. Nach einer Umfrage der Marktforschungsfirma Basisresearch lag Commodore im Bekanntheitsgrad von Computerherstellern hinter IBM und Nixdorf im Dezember 84 auf Platz drei. Zum gleichen Zeitpunkt galt Commodore als der Hersteller, der die preiswertesten und leistungsfähigsten Heim-/Kleincomputer baut und die besten Anwendungsprogramme im privaten Bereich hat. Ein solches Image geht auch nicht mir nichts, Dir nichts verloren — selbst wenn Konkurrenten wie Schneider oder Atari im Lauf dieses Jahres bei derartigen Umfragen aufholen sollten.

Es dürfte auch noch einiges an neuen Programmen, an Peripherie und Zubehör für den 64er auf den Markt kommen — weil seine Leistung für viele Anwendungen ausreicht und weil der Markt so groß ist, daß die Anbieter laufende Entwicklungen nicht »einstampfen« werden. Entwicklungen, die ganz neu in Angriff genommen werden, dürften sich meines Erachtens allerdings eher am 128 orientieren — was sich für die Benutzer erst ab 1986 auswirken dürfte. Der Abstand im Anschaffungspreis dürfte aber nächstes Jahr noch so groß sein, daß der 64er für einen großen Käuferkreis interessant bleibt. Ich glaube auch, daß das Massenprodukt für Commodore interessant genug bleibt, daß es nicht kurzfristig im Trend zu höherwertigen Systemen (PC 10/20; CBM 900) untergeht.

(Michael Pauly, Chefredakteur)

# Hannover-Messe '85

## Neue Besucher- und Ausstellungsrekorde konnte die Hannover-Messe 1985 verbuchen. Viel Neues gab es bei den Computer- und Druckerherstellern zu sehen.

An erster Stelle muß hier natürlich Commodore genannt werden. Der Stand dieser Firma in der CeBIT-Halle 1 war eigentlich nie zu verfehlen. Hinter den größten Menschenansammlungen waren die heißen Produkte kaum auszumachen. Commodore präsentierte gleich drei neue Computersysteme. Den PC 128 (und den Plus/4 übrigens auch) wird es in diesem unserem Lande mit deutscher Tastatur und Umlauten auf dem Bildschirm geben. Commodore scheint sich in der Entwicklung der Computer nun tatsächlich an den nationalen Anwender orientiert zu haben. Für das europäische Ausland soll der PC 128 nämlich mit den jeweiligen Standardtastaturen ausgeliefert werden. Eine erfreuliche Nachricht für Franzosen, Engländer, Dänen und Deutsche. Auch konnten bereits neue Programme für den 128-Modus bewundert werden. Darunter Superscript, ein Textverarbeitungsprogramm mit für diese Preisklasse (198 Mark, von Commodore) außergewöhnlichen Leistungsmerkmalen, wie Textfenstern und Berechnungsmöglichkeiten bei Tabellen innerhalb der Textverarbeitung. Mit Superbase (298 Mark) für den C 128 soll dieses Programm die Daten austauschen können. Der Commodore 128 wird voraussichtlich 1200 Mark kosten (Listenpreis). Ein ausführlicher Testbericht in dieser Ausgabe zeigt Ihnen die Fähigkeiten dieses Computers auf. Als Liefertermin wird Ende Juni oder Juli genannt.

Vom PC 10 (IBM-kompatibler Personal Computer) konnten seit Einführung im Februar nach Aussagen von Commodore bereits 5000 Stück abgesetzt werden. Nun ist auch der PC 20 mit einem 10-MByte-Festplatten-Laufwerk für 7950 Mark erhältlich.

Den Markt der Personal Computer nach oben hin will Commodore mit dem neuen CBM 900 abdecken. Er arbeitet mit dem Betriebssystem Coherent (Unix-kompatibel). Modell 1 dient als Personal Workstation mit Schwerpunkten in CAD (Computer aided design) mit einer Auflösung von 1024 x 800 Punkten, 15 Zoll monochromem Monitor mit 72 MHz, Video-Controller-Karte, Maus, 20 MByte Harddisk und Software für Fenstertechnik. Das Modell 2 ist mehrplatzfähig mit bis zu sieben Terminals. Das Floppy-Laufwerk hat eine Kapazität von 1,2 MByte, der Arbeitsspeicher beträgt 512 KByte RAM (bis 2 MByte ausbaufähig). Mit dieser Modellpalette deckt Commodore wohl als einziger Hersteller das gesamte Leistungsspektrum vom billigen Heimcomputer bis zum mehrplatzfähigen Personal Computer ab.

### Was macht die Konkurrenz?

Als schärfster Konkurrent im Heimcomputermarkt tritt zur Zeit Atari auf. Nachdem Jack Tramiel im Januar 1984 zu Atari wechselte, versucht er mit dieser Firma wieder die Nummer Eins zu werden. Die auf der Hannover-Messe vorgestellten Computer 130 XE und 520 ST lassen einiges erhoffen. Kennzeichnend für Tramiel war von je her die herausfordernde Taktik mit einem überzeugenden Preis-/Leistungsverhältnis und neuer Technologie Marktanteile zu gewinnen (siehe PET, VC 20, C 64). Dadurch zeichnen sich auch die neuen Modelle von Atari aus. Der 130 XE weist 128 KByte RAM aus, ist aber ansonsten mit dem Vorgängermodell 800 XL identisch. Der 520 ST (auch Jackintosh genannt) soll dem Macintosh von Apple das Leben schwer machen. Die Leistungen lassen sich durchaus mit dem »Mac« vergleichen, der Preis ist jedoch ein Hammer, 2795 Mark. Neueste Meldungen, so der Rückzug von Atari von der Consumer Electronics Show in Chicago, gaben allerdings Gerüchten, das Atari (Jack Tramiel) in Schwierigkeiten sei, neuen Nährstoff. Auch die Verschiebung der Auslieferungstermine für die neuen Computer gibt Anlaß zum Nachdenken. Zu wünschen wäre, daß Atari es dennoch schafft, denn mit diesen Computern zu solch einem Preis dürfte einiges im Preisgefüge der Heim- und Personal Computer in Bewegung geraten.

Schneider hat sich mittlerweile zum zweitgrößten Computervertreiber in Deutschland hochgearbeitet. Das Produkt ist gut, der Preis stimmt auch. Entsprechend gut war der Stand besucht.

Apple setzt weiterhin voll auf den Macintosh. Der Fat-Mac mit 512 KByte kostet zirka 8000 Mark.

Sinclair glänzte wieder einmal durch Abwesenheit.

### Die Druckerflut ist im Anrollen

Eine Artenvielfalt sondergleichen gab es auf der Hannover-Messe bei den Druckern zu bewundern. Jeder große Druckerhersteller wartete mit neuen Modellen, nicht zuletzt speziell für den C 64, auf. So kündigte Epson den GX-80 an, ein 9-Nadel-Matrixdrucker mit 100 Zeichen/Sekunde (cps), NLQ-Schrift (Near Letter Quality), 1-KByte-Puffer und natürlich ein auswechselbares ROM-Modul für den C 64. Die Einführung auf dem Markt soll im Juni erfolgen, der Preis wurde mit unter 1000 Mark angegeben.

Seikosha (Vertrieb in Deutschland durch microscan) ist dafür bekannt, daß

fast die gesamte Druckerpalette an alle gängigen Heim- und Personal Computer anschließbar ist. Noch verbleibende Lücken wurden geschlossen. So gibt es den GP-500 und den GP-700 (Farbdrucker) jetzt auch mit dem Zusatz VC was bedeutet, daß diese Drucker direkt an den VC 20, C 16, Plus/4, C 64 und C 128 anschließbar sind. Neu hinzugekommen ist der extrem leise 9-Nadel-Matrixdrucker SP-1000VC mit einer Druckgeschwindigkeit von 100 cps, NLQ (32 x 18 Matrix), einstellbaren Tabulatoren, automatischem Papiereinzug, zehn internationalen Zeichensätzen (und den Commodore-Zeichensatz) und FX-80-Kompatibilität. Der Preis liegt um die 1000 Mark.

Anpassungen an den C 64 hat der Speedy 100-80 (Preis 1099 Mark) von Macrotron erfahren. Komplettiert hat Macrotron die Speedy-Familie. Der Nachwuchs besteht aus dem Speedy 130-80 mit 130 cps und 80 Spalten Breite und den Speedy 130-136 mit 136 Spalten. Die Speedy-Drucker zeichnen sich durch quadratische Nadeln (9x9), multistrike Karbonbänder, standardmäßigem Friktions- und Traktorantrieb sowie elektronisch umstellbare Schriftvarianten aus. Mit dem Speedy thermo gibt es jetzt auch einen billigen Thermodrucker (560 Mark) von Macrotron.

Quen-Data bietet mit dem DMP 1100 VC einen 9-Nadel-Matrixdrucker mit 100 cps, deutschem Zeichensatz und C 64- und Centronics-Interface für 1048 Mark an. Die Schreibmaschine Excellence 70 VC mit eingebautem C 64-Interface kostet 899 Mark.

Star präsentierte zum erstenmal die SG-, SD- und SR-Serie in Europa. Alle diese Drucker verfügen über einen dem Typenradschriftbild angepaßten Modus (NLQ), zudem wurde die Grafikfähigkeit der neuen Generation erweitert. Neben normal-, hoch- und ultrahochauflösender Grafik sind drei zusätzliche Grafikmodi vorhanden: zwei CRT-Modi mit 640 beziehungsweise 720 Punkten pro Zeile sowie ein Plot-Modus mit einer Auflösung von 576 Punkten und gleichem Punktabstand in der Horizontalen und Vertikalen.

Ausführliche Informationen über das Geschehen auf der Hannover-Messe erhalten Sie in der nächsten Ausgabe. (aa)

# Schule braucht Computer

**Es erscheint angesichts der Verbreitung des Heimcomputers in Deutschland schon merkwürdig, daß in einigen Bundesländern der Informatikunterricht noch nicht einmal im Lehrplan aufgenommen ist. Computer in die Schulen zu bringen ist oft immer noch Sache engagierter Schüler, Lehrer oder Eltern.**



Mit diesem Problem hatte auch das Aloisius Kolleg (kurz Ako genannt) in Bonn zu kämpfen. Kein Computer stand den computerbegeisterten Schülern im Unterricht zur Verfügung. Angeregt durch diese Misere und dem Wissen, daß auch in den unteren Klassen viele Jugendliche und Kinder privat mit dem C 64 arbeiteten, entschlossen sich zwei Schüler, den Computerunterricht selbst in die Hand zu nehmen. Zunächst war geplant, mit eigenen, privaten Geräten einen Kurs im Rahmen des Ako-Proseminars (begleitende Kursangebote der Schule nach dem regulären Unterricht) zu starten. Dies hätte bedeutet, daß bei 10 bis 12 Kursteilnehmern (die Meldungen waren wesentlich höher) nur zwei Arbeitsplätze zur Verfügung gestanden hätten. Ein unhaltbarer Zustand. Um die lobenswerte Eigeninitiative der Schüler nicht im Sande verlaufen zu lassen, erklärte sich der VAM (Verein zur beratenden Aufklärung über die Mikrocomputertechnologie e.V.) bereit, kostenlos acht C 64 mit Floppy-Laufwerk und Farbmonitor zur Verfügung zu stellen. Dieser Verein sicherte auch den Service für während des Unterrichts aufgetretene Schäden an den Computern. Die Kurse waren ein voller Erfolg, und die Teilnehmer haben sich alle für einen Fortsetzungskurs angemeldet. Dieses positive Ergebnis bewog dann auch die Schulleitung dazu, sich auf dem Gebiet der Informatik zu engagieren.

Der VAM stellte dafür eigens entwickelte Schulcomputer »Leibniz I« (C 64 mit flacher Tastatur, Floppy-

## Computer ohne Kabelsalat

Ein alter Bekannter im neuen Gewand stellt sich mit dem VAM »Leibniz I« vor. Der C 64 wurde zusammen mit einem Floppy-Laufwerk 1541 und einem Farbmonitor in einem Gehäuse von Knürr untergebracht. Der Monitor ist in allen Richtungen schwenkbar. Die Tastatur ist mit dem Gehäuse durch ein Spiralkabel verbunden. Resettaster an den Seiten für Floppy und Computer runden das Bild ab.

Neue Modelle mit Turboladesystem und Mehrplatzfähigkeit sind in Planung.

Für genauere Informationen über folgende Themen:

* Wie kommen Schulen an den VAM »Leibniz«?
* Gibt es den »Leibniz« auch als Bausatz (kann ich meinen alten C 64 mit Laufwerk und Monitor einschicken und bekomme einen neuen »Leibniz«)?
* Was kann der Leibniz heute, was schafft er morgen?
* Wie ist der Service gewährleistet?

wenden Sie sich bitte an
VAM e.V.
Rathausgasse 30,
5300 Bonn 1

Laufwerk und Farbmonitor in einem Gehäuse integriert, siehe Bild) zur Verfügung. Außerdem wurde ein Computerraum eingerichtet und für das nächste Jahr ein Informatikunterricht geplant.

## Neue Ideen braucht das Land

Dieses kleine Beispiel zeigt deutlich, wie erst durch private Initiative der Computer Eingang in den Schulunterricht findet. Der VAM hat sich dabei Gedanken gemacht, wie für Schulen, die bisher ohne Computer auskommen mußten, weil die Mittel fehlten, der Anreiz geschaffen werden kann, sich eine Anlage zu besorgen. Als Ergebnis kam das Modell der Mietpatenschaften heraus. Dabei werden der Schule durch den VAM die erforderlichen Computer kostenlos zu Verfügung gestellt. Es werden Firmen, Banken, Versicherungen in der Umgebung der Schule gesucht, die auf freiwilliger Basis die monatlichen Mieten für jeweils einen dieser Computer übernehmen. Nach ein paar Jahren sind die Geräte dann »abgezahlt« und gehen in den Besitz der Schule über. Der VAM übernimmt den Service bei anfallenden Reparaturen. So haben alle Seiten etwas davon, die Schüler kommen endlich zum Informatikunterricht, der Schule entstehen keinerlei Kosten und die Firmen der Umgebung können mit Schulabgängern rechnen, die mit hervorragenden Kenntnissen beim Umgang mit Computern ausgestattet sind. Fürwahr eine Idee die Schule machen könnte. (aa)

## Adressenänderung — Oxford-Pascal

In der Ausgabe 5/85 haben wir als Bezugsquelle des neuen Oxford-Pascal CPL Computer plus Soft in Dinslaken angegeben. Diese Firma hat in der Zwischenzeit Konkurs angemeldet und ist nicht mehr lieferfähig. Eine neue Bezugsquelle in Deutschland existiert noch nicht. Interessenten können sich an die Adresse in England wenden:

Oxford Computer Systems (Software) Ltd.
Hensington Road
Woodstock
Oxford OX7 1JR, England

## Competition Pro mit Microschaltern

Der Joystick »Competition Pro« mit Microschaltern wird jetzt von der Dynamics Marketing GmbH angeboten. Erkennbar ist dieser Joystick an dem roten Griff. Er ist im Fachhandel für zirka 72 Mark erhältlich.

## Aktualität vor Kontinuität

Liebe Leser, in letzter Minute haben wir den gesamten Aufbau der 64'er umorganisiert. Es erschien uns wichtiger, Sie mit einem ausführlichen Testbericht über den neuen Commodore 128 zu informieren. Diesem zusätzlichen Platzbedarf sind leider einige Beiträge zum Opfer gefallen. So müssen Sie in dieser Ausgabe auf die beiden Kurse »Assembler ist keine Alchimie, Teil 10« und »Dem Klang auf der Spur, Teil 7« verzichten. Ebenso nicht dabei ist die Anwendung des Monats. All diese Artikel werden in der nächsten Ausgabe wieder zum gewohnt vollständigen Bild des 64'er-Magazins beitragen.

An dieser Stelle möchte ich Sie auch auf zwei Wettbewerbe hinweisen, bei denen insgesamt Preise im Wert von über 25000 Mark zu gewinnen sind. Am Umfragewettbewerb sollte jeder 64'er-Leser teilnehmen, der aktiv an der zukünftigen Gestaltung »seiner« Zeitschrift mitwirken möchte. Ein Personal Computer PC 10 und 10 Disketten-Laufwerke 1541 von Commodore sowie jede Menge Buchgutscheine sind die Belohnung für Ihre Mühen.

Beim zweiten Wettbewerb sind wieder Ihre Programmierkünste gefragt. Schreiben Sie die erste C 64-Telesoftware für Btx. Zu gewinnen sind dabei 2 Btx-Fernseher, eine Reise nach Berlin und zwei MPS-802-Drucker. Mitmachen lohnt sich diesmal also wirklich. (aa)

## Die Kuriositätenecke

Diesmal geht's wieder querfeldein durchs Gemüsebeet mit interessanten, seltsamen und anderen Ereignissen der Computerszene.

—In früheren Zeiten wurden Automatenspiele auf den Computer umgeschrieben, mittlerweile läuft dieser Prozeß schon fast in umgekehrter Richtung ab. Nachdem der Hyper-Olympics-Automat gemäß dem IBM-PC-Vorbild entstand, wurden jetzt auch die Heimcomuterspiele »Boulder Dash«, »Flip & Flop« und »Loderunner« für Automatenhersteller lizenziert.

— Die amerikanische Firma Suncom hat den Vertrieb ihres »Aeorbic Joystick« (richtig gelesen, nicht Aerobic sondern Aeorbic) aufgegeben. Das Gerät wurde auf ein Trainingsfahrrad aufgesteckt. Je schneller man in die Pedale trat, desto schneller bewegte sich das gewünschte Objekt auf dem Bildschirm. Das war gut für Pitstop II, bei Pacman-Fans stellte sich aber ein großer Haßeffekt ein. Auch schien es nicht genug sportliche Heimcomputerbesitzer zu geben.

— Jerry LeMaitre ( 18 Jahre) aus Collingwood in Kanada hat einen Babysitter-Job, den er allerdings mit Hilfe des Computers ausübt. Er zeigt den Kindern, wie man lädt, druckt, Programme startet etc., und in kürzester Zeit haben diese Kinder durch ihren Forscherdrang Basic fast wie von selbst gelernt. Als er seine Lehrkünste an Erwachsenen ausprobieren sollte, hatte er allerdings weniger Erfolg. Sein Kommentar dazu:»They think that the Computer is gonna blow up, when they touch it!« (Sie denken, der Computer explodiert, wenn sie ihn berühren).

— Die Zeitschrift »Computer Games« ist für den Leser wesentlich interessanter, wenn er eine Lupe zur Hand nimmt: Am Rand der Seiten stehen immer wieder kleine Bemerkungen in Miniaturschrift. Der Inhalt: »Catch ya later« oder »Print is dead. We helped to kill it!«.

— Die Anzahl der alternativen Computerzeitschriften Deutschlands nimmt immer mehr zu. Nachdem in München der »Computertrottel« als regionales Blatt schon länger Erfolg hat, versucht sich jetzt eine neue, überregionale Zeitschrift namens »Source« zu etablieren. Themen der Zeitschrift: Allgemeine Informationen ohne Zensur, Programmierkurse mit »Raumschiff Kenterbrise mit Commander Captain Crack«-Szenen, Berichte über das Verhältnis eines seltsamen Völkchens (das man Amerikaner nennt) zu Computern (Computertrottel) und ähnliches mehr.

— Nachruf in einer Mailbox:
In tiefer Trauer geben wir den Tod unserer geliebten NUI DATENKLO z8va8v bekannt.
SIE war uns von ALLEN die Liebste, denn SIE gehörte dem GROEHAZ (GROEssten Hacker Aller Zeiten), Wau Holland vom CCC (Computer Capitalizing Company). Wau durch den »Congress« und den Auftritt bei Frank Elstner finanziell gesundgestoßen, gab nun uns Hackerprolos ca. 3 Wochen DATEX-P aus.
Danke, Wau, große Gesten stehen Dir gut! Weitere Nui-Spenden bitte als allgemeine msgs in dieser Box.
gez. LSD
Liga für subversive Datenkommunikation.

— Eine Studie eines amerikanischen Forschungsinstituts ergab, daß Computerspiele weniger Aggressivität verursachen als Fernsehen oder Darts-Werfen. Dieselbe Studie ergab aber auch, daß Computerspiele die sogenannte »eye-hand-coordination« (Auge-Hand-Koordination) **nicht** verbessern.

— Die Zeitschrift »Electronic games« brachte einen zwei Seiten langen Artikel über mögliche Freizeitbeschäftigungen während des Ladens eines Programms vom 1541-Diskettenlaufwerk. Vorschläge waren zum Beispiel, ein Musikinstrument zu erlernen, ein Buch zu lesen oder gar (während des Ladens von One on One, das immerhin drei volle Minuten benötigt) zum Surfen zu gehen.

— Verschiedene Computerfirmen bekommen ihre Namen auf recht seltsame Art. »Penguin Software« z.B. basiert auf dem »Penguin on the telly«-Sketch der Monty-Python-Gruppe, die in England so etwas ist, wie den Deutschen ihr Otto Waalkes.

— Die University of Montana führte ein wissenschaftliches Experiment durch, in dem Rhesus-Affen das Pac-Man-Spielen beigebracht bekamen. Je besser sie spielten, desto mehr Marshmallow-Früchte bekamen sie. Jetzt sind die Affen so perfekt, daß die Automaten schwerer eingestellt werden mußten.



## RUN = Syntax Error?

*Läßt sich beim C 64 der Basic-Start verändern? Das Vorhandensein der Routine $A68E läßt mich dies zwar vermuten, es gelingt mir aber nicht ganz. Nach dem Heraufsetzen des Basic-Starts wird »RUN« mit »Syntax Error« beantwortet, ebenso »NEW«.* *Germann Podest*

Der Syntax Error bei RUN ist schon ziemlich symptomatisch. Sie haben wahrscheinlich nicht bedacht, daß das erste Byte im Basic-Bereich immer ein Nullbyte sein muß. Vor dem SYS-Befehl müssen Sie also unbedingt einen POKE <Anfangsadresse>, 0-Befehl geben.

## Kreuzworträtsel auf MPS-801?

*Ich habe das fantastische Kreuzworträtsel-Programm von Gert Büttgenbach aus der Ausgabe 12/84 abgetippt. Allerdings ist es mir nicht gelungen, dieses für den Epson-Drucker geschriebene Programm für den MPS-801 umzuschreiben. Vielleicht findet sich ein versierter Leser, der dieses Problem gelöst hat?* *Karl Kipping*

## Chemie per User-Port

*Wie kann man über den User-Port am C 64 chemische Messungen durchführen?*
*Ausgabe: 3/85*
*Joachim von Neuenkirchen*

Die Zeitschrift »CLB — Chemie für Labor und Betrieb« hat seit einiger Zeit eine Rubrik mit Software für den Laborbereich. In einer der letzten Ausgaben erschien ein Artikel über die Soft- und Hardware bei der Erfassung von titrimetrischen Bestimmungen mit einem Commodore 64. Werner Geitner

Info: CLB — Chemie für Labor und Betrieb, Stuttgarter Str. 18-24, Postfach 110262, 6000 Frankfurt 1

### Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

## Teilstrings in Comal?

*Wie kann man in Comal 0.14 Teilstrings analog zu den Basic-Funktionen MID$, LEFT$ und RIGHT$ gewinnen?*
*Detlef Grittner*

Teilstrings werden in Comal 0.14 sehr einfach durch Angabe der linken und rechten Begrenzung erzeugt. Ein Beispiel:

```
DIM TEXT$ OF 20
TEXT$ := "BASIC, COMAL ODER PASCAL?"
PRINT TEXT$ (7:11)
```

Es wird als Antwort der Teil von Textstring vom 7. bis zum 11. Zeichen gedruckt, also »COMAL«. Mit der Vereinbarung L := LEN(X$) werden die Basic-Stringfunktionen in Comal also folgendermaßen erzeugt:

```
LEFT$(X$,N) ... X$(1:N)
RIGHT$(X$,N) ... X$(L-N+1,L)
MID$(X$,N,A) ... X$(N:N+A-1)
MID$(X$,N) ... X$(N:L)
```

## Zufallszahlen mit Wiederholung?

*Beim Arbeiten mit der »Zufallsformel« X=INT(RND(1)*N)+1, durch die Zufallszahlen von 1 bis N erzeugt werden, habe ich festgestellt, daß die so erzeugten »Zufallszahlen« gar keine solchen sind. Zumindestens wiederholen sie sich nach einiger Zeit in gleicher Folge. Nun meine Fragen:*

*Kann man überhaupt von Zufallszahlen reden, wenn sie sich nach einer gewissen Zeit in gleicher Folge wiederholen?*

*Kann man diese Tatsache umgehen, vielleicht durch Änderung der Formel?*

*Wie funktioniert eigentlich die Auswahl der Zufallszahlen?*
*Marco Hesselbart*

Entscheidenden Einfluß auf die erzeugten Zufallszahlen hat die Wahl des oft als unwichtig betrachteten Argumentes der RND-Funktion. Generell sind drei Fälle zu unterscheiden:

Fall 1. Die Funktion wird mit einem positiven Argument aufgerufen, also beispielsweise mit RND(1). Anstelle der 1 kann eine beliebige andere Zahl oder Variable größer als Null stehen. In diesem Falle werden die Zufallszahlen nach einem bestimmten mathematischen Algorithmus, also nach einer bestimmten Rechenvorschrift, ermittelt. In diesem Falle kommen die Zufallszahlen (oder besser die Pseudo-Zufallszahlen), die nach dem Einschalten des Computers erzeugt werden, jedesmal in genau der gleichen Reihenfolge wieder vor, was sicherlich als nachteilig zu bewerten ist. Der Vorteil dabei ist jedoch die sehr gute Gleichverteilung der erzeugten Zahlen. Es werden weder Zahlen bevorzugt noch benachteiligt.




2. Die Funktion wird mit einem negativen Wert aufgerufen (Beispiel: RND(-1) oder RND(X), falls X<0). In diesem Fall ergibt jede einzelne negative Zahl einen speziellen Zufallswert, aber eben für jede Zahl stets wieder denselben. Probieren Sie ruhig zehnmal hintereinander RND(-1) aus, Sie erhalten stets die gleiche Zahl. Haben Sie aber einmal einen als zufällig geltenden Anfangswert X, dann können Sie mit der Folge »10 X=RND(-X): PRINT X:GOTO 10« eine beliebige Kette von zufälligen Zahlen erzeugen. Jeder verschiedene Anfangswert für X liefert bei dieser Methode eine eigene, ganz spezielle Zufallszahlenreihe.

Als Startwert kommen alle möglichen Werte in Betracht, sinnvoll ist in vielen Fällen eine Initialisierung über die Zeitvariable TI. Der Startwert ist dann von der seit dem Einschalten vergangenen Zeit abhängig und kann in den meisten Fällen wohl als ausreichend zufällig angesehen werden.

Fall 3: Die Funktion wird mit dem Argument Null aufgerufen. In diesem Falle wendet der Basic-Interpreter eine weitere Methode an: Er nimmt die Zeiten verschiedener interner Zeitgeber, verknüpft sie durch eine mathematische Formel miteinander und bildet so ebenfalls eine Zufallszahl.

Diese Zahlen sind wirklich alle paar Mikrosekunden völlig anders und liefern keine vorhersehbare Folge von Zahlen. Allerdings ist bei dieser Methode die Gleichverteilung der Zahlen um einiges schlechter, jedenfalls wenn die Zufallszahlen in einer festen Programmschleife, also in einem festgelegten Zeittakt gebildet werden. In diesem Falle werden fast unweigerlich bestimmte Zahlenwerte häufiger vorkommen als andere.

Die effektivste Methode für möglichst zufällige, aber eben auch möglichst gleichverteilte Zahlen ist die zweite. Besonders bei Initialisierung mit TI, womöglich noch durch Multiplikation oder Division mit irgendwelchen Zeropage-Inhalten verknüpft, erreicht man die besten Ergebnisse.

## Adventure-Tips gesucht

*1. Bei dem Adventure »Time Machine« komme ich nicht weiter als bis zum Haus. Alle anderen Wege führen früher oder später in den Treibsand. Wie komme ich ins Haus hinein?*
*2. Wie komme ich bei »Pitfall II« an die Schatulle (es kann auch ein Ring sein)? Ich habe zwar schon einmal alle Schätze, Wölfe, Mäuse, Ratten und Mädchen aufgesammelt, aber an diese Schatulle kam ich bisher einfach nicht heran.*
*3. Wie komme ich bei dem Adventure »The Fabulous Wanda...« an die richtigen Informationen; wo muß ich mein Geld investieren?* *Manfred Wilke*

## Hardcopy mit Hindernissen?

*Das Hardcopyprogramm für Text und Grafik »X-Graf« des CP 80 X-Druckers druckt bei Hardcopies immer eine Leerzeile zwischen den Datenzeilen. Dadurch kann man keine Grafiken ausdrucken. Ich bitte um einen Ratschlag.* *Klemens Millonig*

Stellen Sie an Ihrem Drucker den Auto-Line-Feed-Schalter auf die Aus-Stellung. Näheres darüber steht in Ihrem Drucker-Handbuch.

## Multiplan mit 80 Zeichen?

*Kann man Multiplan auf einem C 64 mit 80-Zeichen-Karte laufen lassen?* *Klaus Grütling*

Nein, das kann man nicht. Multiplan für den C 64 ist für den 40-Zeichen-Bildschirm ausgelegt. Eine 80-Zeichenkarte, die mit Multiplan zusammen läuft, konnten wir außerdem auch nicht entdecken.

# Leser fragen — Willi Brechtl antwortet

## Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

*Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.*

*Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.*

### 3.5-Basic für den C 64?

*Kann man das Commodore 3.5-Basic des C 16 auch in den C 64 »einpflanzen«?*

*Patrick Dohmen*

Direkt aus dem Basic-Interpreter des C 16 kann man die Routinen für diese Befehle nicht übernehmen, da er mit völlig anderer Hardware arbeitet (Betriebssystem und Basic arbeiten mit Bank-Switching, das heißt, es können theoretisch weit mehr als nur 64 KByte Speicher verwaltet werden); eine Übernahme der Befehle ist jedoch schon möglich. Allerdings wird Basic 3.5 noch von keiner einzigen Firma für den C 64 angeboten (aus Copyrightgründen).

Sie müssen sich dafür also selbst eine Befehlserweiterung schreiben.

### C 16-Sprites?

*Ich besitze einen C 16 mit Datasette und wollte wissen, ob ich Sprites im Textmodus simulieren kann oder ob jemand ein Programm dafür geschrieben hat.*

*Frank Karner*

Der C 16 besitzt die Shape- beziehungsweise Sprite-Fähigkeit nur im Hires-Modus, weil im TED (der Chip, der am C 16 auch für die Video-Ausgabe zuständig ist) Sprites nicht hardwaremäßig vorgesehen sind wie beim Videochip des C 64.

Falls man aus Speicherplatzgründen auf Hires-Shapes verzichten will oder muß, kann man sich immer noch, ähnlich wie beim VC 20, mit einem selbstdefinierten Zeichensatz helfen, der nur 2 KByte vom RAM abknappst und dennoch schon beachtliche Grafikmöglichkeiten bietet. Ein entsprechendes Listing dazu mit Zeicheneditor und ausführlicher Beschreibung wie's funktioniert, finden Sie demnächst in dieser Zeitschrift.

### »Magic Desk« gesucht

*In der Ausgabe 7/84 befindet sich ein Bericht über das Programm »Magic Desk«. Da ich großes Interesse an dem Steckmodul habe, bitte ich Sie um Anschrift der Firma, die das Modul vertreibt.*

*Dr. med. Karl Vigano*

Das Programm wurde von Commodore selbst entwickelt und sollte demzufolge eigentlich über jeden Commodore-Händler zu beziehen sein. Da der Programmierer allerdings mit dem ehemaligen Commodore-Chef Jack Tramiel zu Atari gegangen ist, ist nicht sicher, ob Commodore das Modul noch vertreibt. Erkundigen Sie sich am besten bei Ihrem Commodore-Händler, ob der Ihnen das Modul noch besorgen kann.

### Unsichtbar Nachladen?

*Wie schaffe ich es, in Basic ein Nachlade-Programm zu konzipieren, das bei entsprechender Menüwahl ein Teilprogramm nachlädt — und zwar ohne Ausgabe der Meldungen »Searching for« und »Loading«?*

*Ipek Cüneyt*

Beim Nachladen von Programmen müssen Sie folgendes speziell beachten: Der Befehl »LOAD« in einem Programm bewirkt nicht nur, wie im Direktmodus, das Laden eines Programms, sondern gleichzeitig einen Autostart. Falls das nachgeladene Programm größer ist als das ursprüngliche Programm, dann werden alle Variablen gelöscht. Sie sollten daher dafür sorgen, daß Ihr Hauptprogramm immer größer ist als das nachzuladende. Auch sollten Sie beachten, daß Strings in der Regel nicht mit übernommen werden. Wollen Sie Stringvariable dennoch ins nächste Programm übernehmen, dann müssen Sie beispielsweise im ersten Programm schreiben: A$="HALLO"+""

Dies ist deshalb nötig, weil nur Stringvariable, die mit irgendeiner Stringoperation (+, LEFT$, MID$, RIGHT$) verknüpft wurden, ihren Wert auch im nachgeladenen Programm behalten. Um die manchmal unerwünschtes Systemmeldung wie »Searching for« am Bildschirm nicht sichtbar werden zu lassen, gibt es eine sehr einfache Methode: Setzen Sie vor den LOAD-Befehl im Programm einfach die Schreibfarbe auf die Hintergrundfarbe. Sie verhindern damit zwar nicht die Ausgabe dieser Meldung, sie erscheinen aber nicht sichtbar auf dem Bildschirm.

### Atari-Module auf dem C 64?

*Wie bringt man Atari Telespiel-Module (Atari VCS 2600) auf dem C 64 zum Laufen?*

*Robert Wölfert*

Atari-Module bringt man auf dem C 64 überhaupt nicht zum Laufen, da die VCS 2600-Module und der C 64 über eine völlig unterschiedliche Hardware verfügen. Außerdem ist die VCS 2600-Konsole inzwischen schon ab 60 Mark im Ausverkauf erhältlich, so daß sich irgendwelche eventuell möglichen aufwendigen Hardware-Basteleien auch beim besten Willen nicht lohnen.

### Diskette versehentlich formatiert?

*Ich habe versehentlich eine Diskette mit vielen Programmen darauf formatiert. Wie kann ich diese Programme wieder zurückholen?*

*Hinnerk Behn*

Wenn Sie die Diskette ohne Angabe einer ID formatiert haben (die Formatierung kann dann nur einige Sekundenbruchteile gedauert haben), dann ist nur das Directory gelöscht. Mit einem Diskettenmonitor (und unserem Floppy-Kurs) könnten Sie bei etwas Erfahrung in der Lage sein, das Inhaltsverzeichnis wieder zu restaurieren. Da in der Regel aber mit ID formatiert wird, sind Ihre Programme wohl verloren. Das 1541-Laufwerk beschreibt nämlich beim normalen Formatieren alle Sektoren der Diskette mit Null Bytes. Die ursprünglich darin enthaltene Information geht dabei genauso verloren wie beim Überspielen einer normalen Musik-Kassette.

### Timing-Probleme?

*Wenn ich eine selbstprogrammierte Melodie ständig wiederholen möchte, indem ich hinter die DATA-Zeilen einen RESTORE-Befehl schreibe und mit GOTO wieder an den Anfang springe, dann hat der Computer einige Timing-Schwierigkeiten, besonders bei schnellen Melodien.*

*Frank Herbert*

Abhilfe: Lesen Sie die Musikdaten zu Anfang in eine Feldvariable ein (Dimensionierung nicht vergessen). Der Zugriff auf Feldelemente ist schneller als auf DATAs. Außerdem sollten Sie anstelle des GOTO-Befehls den entsprechenden Programmabschnitt durch eien FOR...NEXT-Schleife wiederholen lassen, auch das bringt einen Geschwindigkeitsvorteil. Und natürlich sollten Sie alle Zahlenwerte schon in der Form »Low-Byte, High-Byte« im Programm stehen haben und die Berechnungen nicht dem Computer während der Programmausführung aufbürden.

### Zeichensatz verschwunden?

*Aus welchem Speicherblock kann man die Daten eines geänderten Zeichensatzes PEEKen, wenn der Zeichensatz nach $E000 und das Video-RAM nach $CC00 verlegt worden ist? Ich habe den ganzen Speicher durchsucht, aber nichts gefunden.*

*Jens Schlingmann*

Ab Adresse $E000 sind die 8 KByte ROM des Betriebssystems dem RAM-Bereich überlagert. Während man mit POKE auch in diesem Bereich ins »unter« dem ROM liegende RAM schreiben kann, ergibt jedes PEEKen hier nur den Inhalt des Betriebssystem-ROMs. Abhilfe ist hier nur über Maschinensprache möglich, da zum Auslesen dieses Speicherbereichs das Betriebssystem und der Basic-Interpreter abgeschaltet werden muß.

### MSE-Probleme?

*Es gelingt mir nicht, mit dem MSE eingegebene Programme nach dem Abspeichern und Wiederladen laufen zu lassen. Es ist überhaupt kein Programm mehr vorhanden: Nach LIST erscheint sofort »Ready«, ebenso nach RUN.*

*Stafan Gass*

Der MSE dient zur Eingabe von Programmen, die in Maschinensprache geschrieben sind. Maschinenprogramme lassen sich von Basic aus nicht listen und müssen mit dem SYS-Befehl statt mit RUN gestartet werden. Allerdings muß man dazu die Startadresse kennen. Diese Startadresse müssen Sie der Anleitung zum Programm entnehmen. Lesen Sie sich bitte zu jedem Maschinenprogramm die Anleitung sehr genau durch.

## Druckeranschluß ohne Software?

*Ich besitze seit kurzer Zeit einen VC 20 und den Drucker GP 80A von Seikosha. Wie kann ich diesen Drucker an den VC 20 anschließen, ohne daß ich ein Programm benötige?*

*Wilhelm Halla*

Den GP80 gibt es in einer speziellen Version zum direkten Anschluß an den VC 20 als GP 80 VC von Seikosha oder als VC 1515 im Commodore-Vertrieb. Bei Ihrer Version (A) handelt es sich um das Standardmodell mit internationalem (ASCII-) Zeichensatz und Centronics-Schnittstelle. Sie benötigen zum Anschluß entweder ein VC-Interface (seriell auf Centronics) oder einfach ein Centronics-Kabel zum Anschluß an den User-Port. Ein VC-Interface ist allerdings nicht ganz preiswert (etwa von 100 Mark aufwärts), während ein Centronics-Kabel nur ein paar Mark kostet, dafür aber ohne zusätzliche Software nicht zu betreiben ist. Ihr Seikosha-Fachhändler kann Ihnen nähere Auskünfte geben.

## Steckmodul abschalten?

*Ich besitze einen C 64 und seit einiger Zeit ein Steckmodul für die Spracherweiterung Simons Basic. Leider laufen einige Programme nicht mit dieser Erweiterung. Durch ständigen Wechsel des Moduls sind bereits Leiterbahnen auf der C 64-Platine beschädigt worden.*

*Ist es möglich, durch Einbau eines Schalters in die Leiterbahn des Pin 2 am Expansion Port (+5V) das Modul abzuschalten ohne eine Beschädigung herbeizuführen?*

*Dieter Kothe*

Eine entsprechende Bastelanleitung erscheint demnächst unter der Rubrik Tips & Tricks.

## Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene, dann schreiben Sie uns. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

## Merge-Problem

*Wie kann ich zwei Programme im Speicher zu einem einzigen zusammenfügen?*
*Ausgabe 4/85*

*Martin Hossdorf*

Auch ohne Basic-Erweiterung kann man beim C 64/VC 20 Programme relativ einfach »mergen«:

Erstes Programm laden und danach »POKE 43, PEEK(45)-2 : POKE 44, PEEK(46)« eingeben. Dies schützt den ersten Programmteil. Jetzt kann man ein zweites Programm laden, nochmals die gleichen POKE-Befehle geben, ein drittes Programm laden und so fort. Nach »POKE 43,1 : POKE 44,8« stehen alle Programme zu einem einzigen zusammengefaßt im Speicher. Man sollte allerdings darauf achten, daß die nachgeladenen Programme höhere Zeilennummern haben, da es sonst zu Problemen kommen kann.

Michael Aßfahl

## Schreibmaschine am Computer lernen

*Wo gibt es ein Programm, das Schreibmaschinenkenntnisse am Bildschirm vermittelt?*
*Ausgabe 4/85*

*Christian van der Linde*

Ich habe einen Schreibmaschinenkurs für den Commodore 64 entwickelt. Das Programm wurde auf der Hannover-Messe 1984 von Bundesforschungsminister Dr. Riesenhuber als bestes Lernprogramm mit der »Goldenen Diskette« ausgezeichnet.

Der Schreibmaschinenkurs ist in deutscher Sprache geschrieben und führt in 27 Lektionen auf unterhaltsame Weise in das Zehn-Finger-System ein. Es verwendet den normalen C 64-Zeichensatz. Einige grafische Einlagen lockern den Lernprozeß auf und sorgen für notwendige (Entspannungs-)Pausen.

Das Programm ist in der Reihe Chip-Soft des Vogel-Verlags auf Diskette und Kassette erschienen.

Nils Rode

Info: Chip-Soft, Vogel-Verlag, Postfach 6740, 8700 Würzburg 1

## Nochmals Simons Basic und Turbo Tape

*Wer kennt eine Möglichkeit, um gleichzeitig Simons Basic und Turbo Tape benutzen zu können?*
*Ausgabe 8/84, 12/84, 2/85*

*Rolf Lehr*

Dieses Problem ist eigentlich schon seit Mitte 1984 gelöst. Das zu Turbo Tape kompatible Fast Tape von H. Reichert gestattet auch bei Simons Basic eine schnelle Kassette. Interessenten mögen sich bitte bei mir melden.

Jürgen Atrott

In der amerikanischen Computerzeitschrift »Compute!« wurde im Januar ein Turbo Tape zum Abtippen veröffentlicht, das man ans jeweilige Ende des Basic-RAM setzen kann. Dieses Turbo Tape ist kompatibel mit Simons Basic und anderen Erweiterungen. Ich bin gerne bereit, Interessenten gegen Einsendung einer Leerkassette und eines frankierten Rückumschlages eine Kopie zu senden.

Marcus Maier

## Fortran Compiler für C 64

*Gibt es für den C 64 einen Fortran-Compiler?*
*Ausgabe 3/85*

*Bernhard Pöttinger*

Commodore selbst bietet in Amerika einen Fortran Compiler für den C 64 mit CP/M-Modul an. Dieser Compiler (»Nevada Fortran«) ist jedoch leider in Europa nicht erhältlich. Es handelt sich dabei um eine bemerkenswert starke Fortran IV-Version mit ausgezeichneter Dokumentation.

Tom V. Segalstad

Info: Commodore Business Machines, Inc., 1200 Wilson Drive, West Chester, PA 19380

## Mikrofon am C 64

*Wie kann ich ein Mikrofon an den C 64 anschließen, und wie läßt sich das Tonsignal dann beeinflussen?*
*Ausgabe 3/85*

*Roland Feichtinger*

Der Anschluß eines Mikrofons an den C 64 ist über den Audio-In-Pin an der Audio-Video-Buchse möglich. Das Signal muß allerdings vorverstärkt werden. Dies geht am besten über die heimische Stereo-Anlage. Man schließt also das Mikrofon an den entsprechenden Eingang der Stereo-Anlage an und verbindet den Kopfhörer-Ausgang mit Audio-In am C 64. Natürlich darf auch die Masseleitung nicht vergessen werden (GND). Die Pinbelegung von C 64 und Stereoanlage entnimmt man dem C 64-Handbuch beziehungsweise dem Schaltplan der Stereoanlage.

Achtung: Bei nicht sachgemäßer Ausführung der Verbindung können sowohl der C 64 als auch die Stereoanlage beschädigt werden. Elektronik-Laien sollten daher vom Nachbau absehen. Und noch eine Warnung: Hüten Sie sich davor, den Verstärker zu weit aufzudrehen. Durch zu hohen Eingangspegel kann der SID im C 64 zerstört werden.

Der Ton vom Mikrofon kann durch POKEn der Lautstärke hörbar gemacht werden. Weiterhin stehen alle Filtermöglichkeiten des SID zur Verfügung.

Holger Riedelbauch

## Schreibmaschine am C 64

*Wie kann ich eine Olivetti Praxis 30 an den C 64 anschließen?*
*Ausgabe 8/84 und 3/85*

*Josef Hauk*

Leider enthält die in Ausgabe 3/85 gegebene Übersicht über die für die Brother CE-Reihe (50, 51, 60, 70) erhältlichen Interfaces eine interessante Variante nicht, und zwar das Modul Robotext der Schweizer Firma Märki & Lenz. Robotext ist ein komplettes Textsystem (Textverarbeitung, Adressenverwaltung, Interface) und wird in der BRD durch C.M. Hacker Consult vertrieben.

Dr.-Ing. Winfried Staib

Der Anschluß der Brother CE-50 bis CE-70 sowie der Olivetti P-30 bis P-40 ist mit dem Tri-Interface von Märki und Lenz direkt und ohne Probleme möglich. Außerdem ist ein EPROM erhältlich, mit dem das Interface voll zu Vizawrite kompatibel wird.

Alfred Theiler

Info: Märki, Lenz & Co., Bernstr. 15, CH-3114 Wichtrach
C.M. Hacker Consult, Paulinenstraße 4, 7107 Neckarsulm

## Software für Lichtgriffel gesucht

*Wer hat ein Lichtgriffel-Programm für den VC 20, mit dem man auch in hochauflösender Grafik zeichnen kann?*

*Andreas Hattemer*

## Elektrische Schreibmaschine am C 64

*Ich suche Bausätze, Anleitungen, Soft- und Hardware zum Anschluß einer alten elektrischen Triumph Schreibmaschine an den C 64.*

*Hellmut Korndörfer*

## Q-Bert für Grundversion

Das in der Ausgabe 2/85 veröffentlichte Spiel Q-Bert für den VC 20 gefällt mir ausgezeichnet. Leider wurde vergessen zu erwähnen, daß es nur mit der 3-KByte-Erweiterung läuft. Es ist mir nun gelungen, das Programm so weit zu komprimieren, daß es auch in die Grundversion paßt. Gegen Erstattung der Unkosten für Porto und Datenträger sende ich das Programm gerne jedem Interessierten zu.

Fred Behringer

## Hier gibt's Comal-Software

Der Comal-User-Club C. Canisius bietet seit Anfang Mai in Comal geschriebene »Public Domain« Software für den C 64 auf Diskette an. Jede Diskette ist doppelseitig bespielt, kostet 25 Mark und enthält in der Regel 20 bis 40 Programme.

Christiane Canisius

Info: Comal-User-Club C. Canisius, Freiheitsstr. 30, 4000 Düsseldorf 12


G4ER ONLINE

# Erster ausführlicher Test PC 128 (Teil 1)

**Der PC 128 war bei Commodore noch im Vorbereitungsstadium für den deutschen Markt, da hat die 64'er-Redaktion sich eines der ersten Geräte über die Osterfeiertage besorgt, um den PC 128 für Sie auf Herz und Nieren zu testen. In diesem Teil beschreiben wir ausführlich die Hardware, das Basic 7.0 und die Kompatibilität zum C 64.**

Nicht nur die 6510-Programmmierer kommen mit dem PC 128 auf ihre Kosten. Der PC 128 hat nämlich zusätzlich einen Z80-Prozessor eingebaut, der einen wesentlich komfortableren Befehlssatz als die 65xx-Familie besitzt, zu der auch der 8502-Prozessor im PC 128 gerechnet werden muß. Der Z80 wird im PC 128 mit 4 MHz getaktet. Dadurch, und mit dem neuen 1571-Laufwerk, das auch CP/M-Disketten lesen kann, werden endlich CP/M-Programme wie Wordstar auf einem preiswerten Computer verwendbar.

Der PC 128 unterscheidet sich vom C 64 in folgenden Punkten:

**1. Geschwindigkeit**

Der 8502-Prozessor des PC 128 kann mit zwei wählbaren Frequenzen getaktet werden; mit 1 MHz oder mit 2 MHz. Zur Umschaltung braucht in einem Basic-Programm nur der SLOW-(1 MHz) oder der FAST-(2 MHz-)Befehl gegeben werden.

## Slow und Fast

Im SLOW-Modus werden Programme etwas langsamer als beim C 64 (auch 1 MHz) verarbeitet. Das kommt durch die komplizierte Verwaltung des 128 KByte-Speichers des PC 128 zustande. Diese Verlangsamung ist aber kein Grund, den PC 128 als lahmen Computer zu verdammen, hat man doch sehr leistungsfähige Basic-Befehle zur Verfügung. Und gerade diese können den PC 128 schneller als den C 64 machen. Denn man muß nicht mehr mit Unmengen von POKE-, PEEK- und SYS-Befehlen arbeiten, die in einem Programm viel Zeit »fressen« können. Der Prozessor braucht nun mal wesentlich mehr Taktzyklen, wenn beispielsweise der Grafik-Modus mit POKEs, anstelle mit einem einzigen Maschinenprogramm eingeschaltet wird. Denn jeder einzelne POKE-Befehl muß erst vom Basic-Interpreter erkannt und ausgewertet werden, bevor an eine Ausführung zu denken ist.

Der FAST-Modus läßt jedes Programm um das Doppelte schneller werden. Mit einem Haken allerdings: Es ist ein RGB- oder Schwarz-Weiß-Monitor erforderlich.

Während für den SLOW-Modus der Video-Interface-Chip (VIC) aus dem C 64 noch ausreicht, muß im FAST-Modus ein anderer, schnellerer Baustein verwendet werden, denn der VIC aus dem C 64 »schafft« maximal 1 MHz Taktfrequenz. Der neue Video-Controller hat die Bezeichnung 8563. Er übernimmt im FAST- und 80-Zeichen-Modus die Bilderzeugung. Im Unterschied zum VIC (er liefert ein Composite-Videosignal) stellt der 8563 leider nur ein genormtes RGB-(IBM-kompatibel) und Luminanz-(Helligkeits-)Signal bereit.

Die RGB-Norm sieht für jede Grundfarbe ein eigenes Intensitäts-(Helligkeits-)Signal vor. Dadurch wird eine wesentlich größere Bildauflösung (Punktschärfe) als bei der Composite-Norm möglich. Die Composite-Norm kennt nur zwei Signale: ein Farb- und ein Luminanzsignal.

Für den Benutzer bedeuten die zwei verwendeten Normen, daß er zwei Monitore braucht. Einen Composite-Monitor für den SLOW- und C 64-Modus (ein Fernseher reicht notfalls auch) und einen RGB- oder Schwarzweiß-Monitor (mit Luminanzeingang) für den FAST- und 80-Zeichen-Modus.

Eine einfachere Lösung bietet der neue für den PC 128 lieferbare Commodore-Monitor 1902, der mit einem Schalter von RGB auf Composite umgeschaltet werden kann.

**40/80-Zeichendarstellung und Grafik**

Wie bei Personal Computern schon immer üblich, kann der PC 128 auch achtzig Zeichen darstellen. Allerdings nur auf einem RGB-Monitor; unabhängig vom SLOW- oder FAST-Modus.

Ein großer Nachteil ist allerdings der fehlende Grafik-Modus bei FAST oder 80-Zeichen-Darstellung. Der dazu benötigte Video-Controller kann nämlich nur 80 Zeichen (und keine Hires-Grafik!) in einer Auflösung von 640 x 200 Punkten in 16 Farben darstellen und sonst nichts. Ein Zeichen besteht dabei aus 8 x 8 Punkten, Zeilen- und Zeichenabstand mitgerechnet. Hochauflösende Grafik, Sprites (bewegbare Objekte, die selbst definiert werden können) und zweifarbige Buchstaben sind in dieser Auflösung (620 x 200 Punkte) nicht möglich. Grafikbefehle wie DRAW, CIRCLE und BOX sind im 80-Zeichen-Modus ebenfalls nicht funktionsfähig, da eben der Video-Controller keinen eigenen Grafikprozessor wie der Video-Interface-Chip hat. Ein Grafik-Modus existiert bei 2 MHz also nicht.

Im 40-Zeichen- und SLOW-Modus übernimmt der aus dem C 64 bekannte Video-Interface-Chip (VIC) das Regiment. Für den PC 128 wurde er überarbeitet und heißt 8564. Das bedeutet, daß die grafischen Fähigkeiten des 40-Zeichen- und

### In letzter Minute

haben wir den Erlkönig für den deutschen Markt zu Gesicht bekommen. Der »deutsche« PC 128 überrascht mit einem kleinen, aber sehr wichtigen Detail: er verfügt über eine deutsche Tastatur und kann auch Umlaute auf dem Bildschirm darstellen. Mehr darüber in der nächsten Ausgabe.



SLOW-Modus denen des C 64 entsprechen.

Wir haben teilweise mit zwei gleichzeitig angeschlossenen Monitoren gearbeitet, einem RGB-Farbmonitor und einem Commodore 1701. Es war schon ungewöhnlich, am 80-Zeichen-RGB-Bildschirm Grafik-Befehle zu programmieren und zu starten und auf dem 1701 (Composite-Monitor) den Grafikaufbau zu beobachten. Schaltet man nämlich im 80-Zeichen-Modus auf Grafik-Modus, wird der für die Darstellung von 80 Zeichen benötigte 8563-Video-Controller abgeschaltet und dafür der 8564-VIC mit seinem Grafikprozessor aktiviert. Alle Bildausgaben gehen dann über den Composite-Ausgang. Da der 8563-Video-Controller dann zwar nicht mehr vom Prozessor angesprochen wird, das zuletzt gesendete RGB-Bild aber nicht gelöscht wird, bleibt auf dem RGB-Monitor das 80-Zeichen-Textbild weiterhin zu sehen.

## Zwei in Einem

Wie anfangs erwähnt: Der PC 128 besteht aus zwei oder eigentlich aus drei Computern. Einem C 64, einem PC 128 und einem CP/M-Computer. Für diese drei Betriebsarten stehen zwei Prozessoren zur Verfügung: ein 8502 und ein Z80. Der 8502 ist vollständig kompatibel zu dem 6510 im C 64, kann aber mit 2 MHz getaktet werden. Der Zilog Z80 bedarf eigentlich keiner Erklärung. Er wird seit vielen Jahren in vielen großen und kleinen Computern eingesetzt, sowohl im ZX-81 als auch in großen CP/M-Maschinen.

Schon beim Einschalten des PC 128 macht sich der Z80 bemerkbar. Er versucht das CP/M-Betriebssystem von Diskette zu booten (zu laden und zu starten). Ohne irdendeinen Befehl beginnt dabei das angeschlossene Diskettenlaufwerk zu laufen und versucht das Programm zu laden. Wird kein entsprechendes Programm gefunden, aktiviert der Z80 den 8502-Prozessor und der PC 128-Modus wird eingeschaltet.

## Geteilte Datenschienen

Beide Prozessoren, Z80 und 8502, sind in der Lage miteinander zu kommunizieren, was auch im Betriebssystem vorgesehen ist. Reicht nämlich für bestimmte I/O-Operationen der BIOS-(Betriebssystem-)Befehlssatz der CP/M nicht aus, übernimmt der 8502 diese Aufgaben. Wie weit man sich das zunutze machen kann, bleibt abzuwarten. Wir werden genaueres über den Z80 und das CP/M-Betriebssystem in der nächsten Ausgabe berichten.

Z80 und 8502 teilen sich im PC 128 die Adreß- und Datenleitungen. Da der Z80 schneller arbeitet als die übrigen Bausteine, paßt eine Interface-Schaltung die Geschwindigkeit des Z80 an das System an, was natürlich die Arbeitsgeschwindigkeit des Z80 verringert. Diese Interface-Schaltung sorgt dafür, daß der Z80 bei Buszugriffen nur mit 2 MHz anstelle der angegeben 4 MHz getaktet wird. Das bedeutet, daß CP/M-Programme auf dem PC 128 nicht ganz so schnell laufen, wie auf einem 4 MHz-CP/M-Computer.

### Schneller 8502

Der auffälligste Unterschied des 8502 zum 6510 im C 64 liegt in seiner Geschwindigkeit. Der 6510 »verträgt« nämlich nur Taktfrequenzen bis 1 MHz, der 8502 aber bis zu 2 MHz. Der 8502 kann also doppelt so schnell arbeiten. Jedes Programm, sei es Basic oder Maschinensprache, kann demnach auf dem 8502 doppelt so schnell gefahren werden wie auf dem 6510.

### PC 128 – der Grafikbeschleuniger?

Diese Frage ist nur mit »Jein« zu beantworten. Einerseits schaltet, wie erwähnt, der VIC im Fast-Modus einfach ab. Der Bildschirm sieht dann genauso aus, wie beim Laden von Datasette. Andererseits werden alle Grafikbefehle im Speicher ausgeführt, so daß nach Zurückschalten auf SLOW die fertige Grafik sichtbar wird. Das gleiche gilt generell, wenn Sie nur einen Fernseher oder 1701/1702-Composite-Monitor besitzen und trotzdem mit 2 MHz Taktfrequenz arbeiten möchten. Sie müssen vor jeder Bildschirmausgabe mit dem SLOW-Befehl den 1-MHz-Modus einschalten, denn der VIC ist ja bei 2 MHz abgeschaltet. Erst nach der Umschaltung kann der VIC dann wieder Ihren Fernseher oder 1701/1702 ansteuern. Nach der Ausgabe können Sie dann mit FAST wieder den 2-MHz-Takt aktivieren. Das Programm läuft dann wieder mit doppelter Geschwindigkeit — allerdings mit abgeschalteten Composite-Bildschirm, da bei 2 MHz wieder der »simple« RGB-Video-Controller das Bild erzeugt.



**Bild 4. Der PC 128 von hinten gesehen. Von rechts nach links: User-Port, RGB-Ausgang, Fernseher, Composite Video, Serieller Port, Datasetten-Anschluß, Expansion-Port für Steckmodule**



**Bild 7. Anschlüsse und Schalter an der rechten Seite: Netzteilanschluß, Einschaltknopf, Reset-Taster, Joystick-Ports 2 und 1**

Bild 1 zeigt die Speicherorganisationen der möglichen Betriebsarten. Die Speicheraufteilung im C 64-Modus ist mit der Memory Map des C 64 vollkommen identisch: im Bereich von $A000 bis $BFFF liegt das Basic-ROM und von $E000 bis $FFFF das Kernal-ROM mit dem 40-Zeichen-Editor. $D000 bis $DFFF belegen die I/O-Bausteine (CIA und VIA) und das Character-ROM. Beide ROM-Bereiche, Basic und Betriebssystem für den C 64-Modus, sind beim PC 128 in einem 16-KByte-ROM-Baustein abgelegt, im Gegensatz zum C 64, wo Basic und Betriebssystem in je einem 8-KByte-ROM untergebracht sind.

## Die Speicherlandschaft

Im PC 128-Modus wird die Aufteilung komplizierter; ist doch ein wesentlich umfangreichers Basic (Version 7.0) und ein zusätzlicher 80-Zeichen-Editor unterzubringen.

Das Basic-ROM gliedert sich in zwei Teile á 16-KByte: Basic Teil 1 und Teil 2 in der Memory Map (Bild 1 und 2). Der erste Teil liegt zwischen den Adressen $4000 und $7FFF, der zweite geht von $8000 bis $BFFF. Insgesamt sind das 32 KByte Basic-ROM. Zum Vergleich: die Basic-Version 2.0 des C 64 ist nur 8 KByte lang.

Das Betriebssystem (Kernal) des PC 128 befindet sich zwischen den Adressen $E000 und $FFFF, wie beim C 64. Das Betriebssystem enthält ein Monitorprogramm. Die Routinen für den 40/80-Zeicheneditor liegen von $C000 bis $CFFF.

Für Basic-Programme hat der PC 128 etwa 81,5 KByte mehr Platz als der C 64. Möglich wird das durch Bank-Switching zwischen zwei 64 KByte großen RAM-Bänken.

#### Was ist Bank-Switching?

Es ist nicht möglich, mit einem Prozessor, der 16 Adreßleitungen hat, wie alle 65xx und der 8502, mehr als $2^{16}$ = 65536 Speicherzellen (64 KByte) direkt zu adressieren. Will man mehr Speicher als 64 KByte »haben«, gibt es nur eine Möglichkeit, will man keinen leistungsfähigeren Prozessor verwenden: Das Bank-Switching.

Bank-Switching heißt soviel wie Speicherblock-Umschaltung.

Die 128 KByte Speicher des PC 128 werden dazu in zwei Teile mit je 64 KByte gespalten. Mit einem Trick wird dafür gesorgt, daß der Prozessor abwechselnd die eine oder die andere 64-KByte-Bank »sieht«. Der Trick heißt Memory Managment Unit (MMU). Wie der Name schon sagt, managt diese Schaltung die Speicherkonfiguration. Die MMU bestimmt, auf welche RAM-Bank der Prozessor »sehen« darf, also wo Schreib-/Lesezugriffe im Speicher erfolgen sollen. Aber nicht nur das. Die MMU gibt auch die ROM-Konfiguration an, sie sagt also dem Prozessor, aus welchem ROM er seine Befehle zu holen hat. Entweder aus dem Kernal oder aus einem EPROM einer Erweiterungskarte.

Die MMU ist kein toter Baustein, an dem nichts verändert werden kann. Im Gegenteil. Bei unserem Test kamen wir auf die interessantesten Ideen, die man mit diesem Baustein realisieren könnte, angefangen von einem Interrupt-gesteuerten Kopierschutz, der sich auf verschiedene Bänke verteilt, bis zu mehreren Programmen, die auf verschiedenen Bänken »sitzen«.

Im PC 128 wird der Basic-Speicher so verwaltet, daß die RAM-Bank 0 (64 KByte) für Basic-Programme und Bank 1 (64 KByte) für Basic-Variable reserviert ist. Für den Basic-Programmierer bedeutet das, daß er je etwa 60 KByte Speicher für das Programm und die Variablen zur Verfügung hat. Es nicht möglich, größere Programme auf Kosten des Variablenspeichers anzulegen. Die vollen 64 KByte pro Bank können auch nicht vollständig genutzt werden, da ein Bereich in beiden Bänken für die Zeropage, den Stack und den Bildschirmspeicher reserviert ist. Der Bereich geht bis $0400. Während ein Basic-Programm läuft, regelt die Memory Managment Unit, auf welche Bank zugegriffen werden soll. Die Informationen darüber, ob gerade eine Variable oder Befehle zu verarbeiten sind, erhält die MMU vom Basic-Interpreter.

## 122365 Basic Bytes Free

Aber nochmal zurück zu dem Bereich von $0000 bis $0400 der für das System reserviert ist. Die Besonderheit daran ist, daß in diesem Bereich nur Bank 0 existiert. Bank 1 kann dort nicht angesprochen werden. Auf diese Weise ist sichergestellt, daß der Prozessor immer auf dieselbe Bank zugreift und nicht deshalb abstürzt, weil in Bank 1 vielleicht ein anderer Stack steht als in Bank 0.

Zusätzlich kann man sich selbst Bereiche von 1 bis 16 KByte in beiden Bänken reservieren, die am Speicheranfang oder Speicherende liegen können (Bild 2).

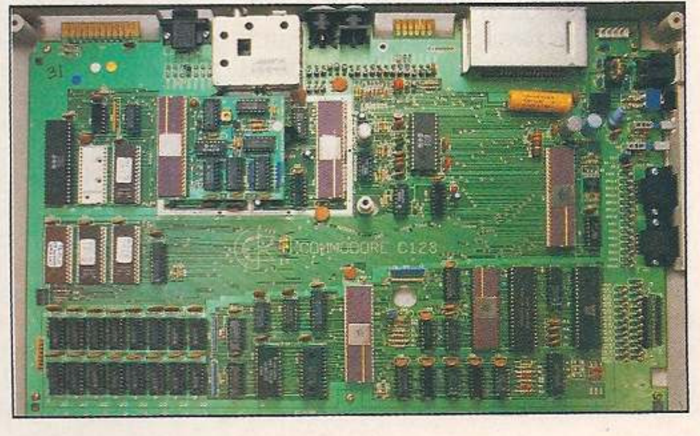


**Bild 6. Ein Blick auf die Platine des PC 128. Für das Seriengerät sind allerdings noch einige Änderungen vorgesehen.**

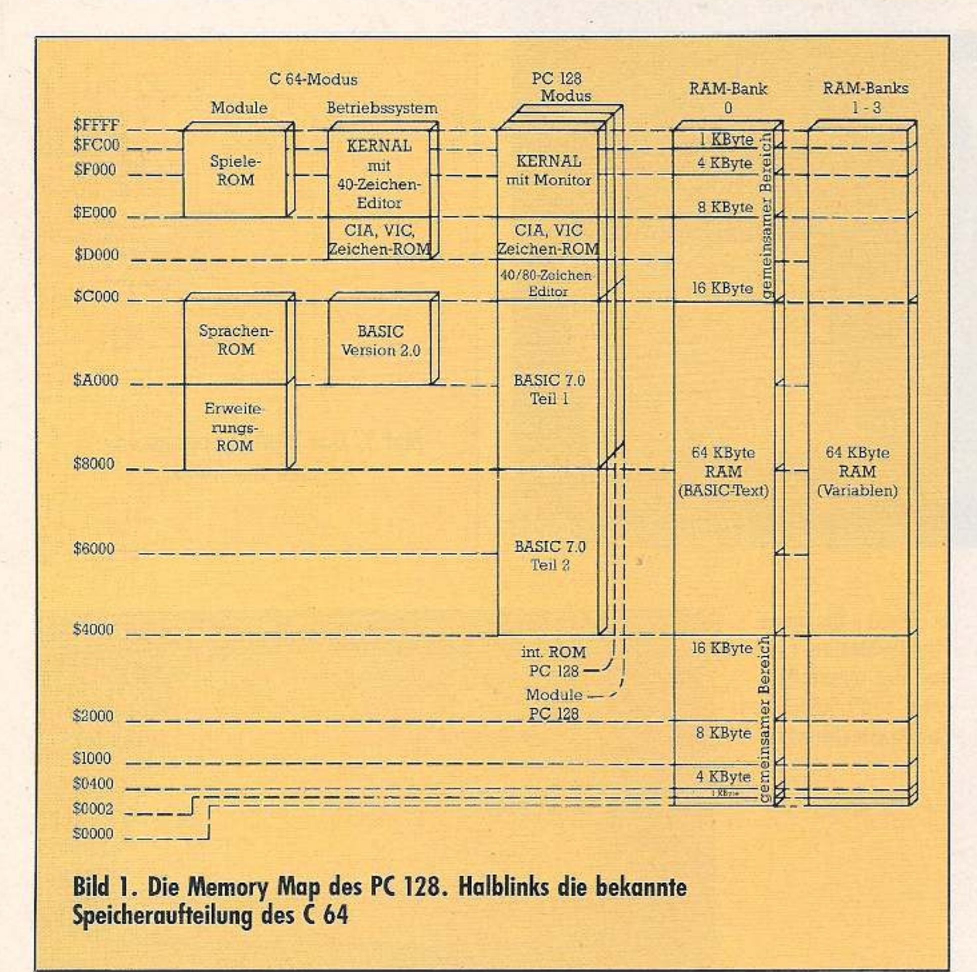



**Bild 1. Die Memory Map des PC 128. Halblinks die bekannte Speicheraufteilung des C 64**

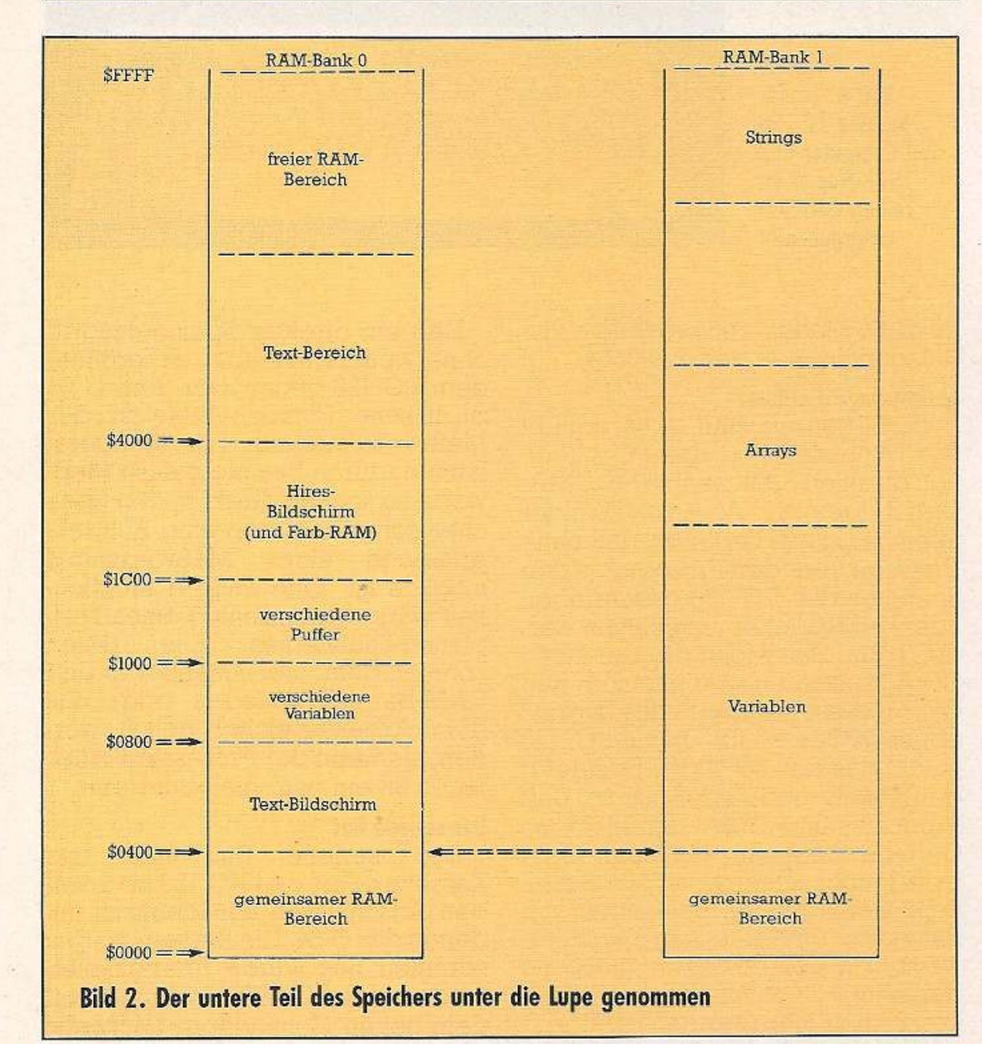



**Bild 2. Der untere Teil des Speichers unter die Lupe genommen**

Man kann diesen Bereich beispielsweise in eigenen Maschinenroutinen als Stack oder Speicher verwenden, wenn zwischen den Bänken umgeschaltet werden muß und die gleichen Daten zur Verfügung stehen sollen.

Zur Bereichswahl hat die MMU das RAM-Configuration-Register (RCR, Bild 3). Bit 0 und 1 des RCR bestimmen die Größe des gemeinsamen Speicherbereichs von Bank 0 und Bank 1: 1, 4, 8, oder 16 KByte. Sind beide Bits »0«, beträgt der gemeinsame Speicher 1 KByte, oder wenn beide »1« sind, 16 KByte.

Bit 2 und 3 des RCR bestimmen die Bereichslage. Ist Bit 2 gesetzt, liegt der gemeinsame Bereich an der Speicheruntergrenze, ist Bit 3 gesetzt liegt er an der Speicherobergrenze. Wenn beide Bits »1«, gesetzt sind, wird sowohl am Anfang und am Ende des Speichers der angegebene Bereich reserviert. Der Bereich kann also entsprechend zum jeweiligen Programm angelegt werden. Wie schon erwähnt, liegt der gemeinsame Speicher immer in Bank 0. Aber die MMU weiß durch das RCR, bei welcher Adresse die Bank gewechselt werden soll, ohne daß der Wechsel explizit im Programm angegeben werden muß.

## Der Organisator — Die Memory Managment Unit

Die MMU regelt den Aufbau des Speichers. Sie bestimmt, welche RAM-Bank aktiv ist oder beim nächsten Zyklus aktiviert werden soll. Die interessantesten Register der MMU sind das schon erwähnte RAM Configuration Register (RCR, Bild 3) und das Configuration Register (CR). Das CR kontrolliert die ROM-, die RAM- und die I/O-Konfiguration des PC 128. Das Register hat die Adresse $D500 im I/O- und $FF00 im Kernal-Bereich. Das CR bei $D500 wird nur bei I/O-Zugriffen benötigt. Die MMU stellt sich dann das Register selbst ein. Findet kein I/O-Zugriff statt, ist das CR, mit den gesamten I/O-Routinen, in der Memory Map nicht vorhanden. Im Gegensatz zum CR bei $FF00, das ständig in der Memory Map präsent ist.

Bit 0 des Configuration Register (CR) regelt im PC 128-Modus den Prozessorzugriff; entweder auf den I/O-Bereich ($D000-$DFFF, High) oder auf das ROM/RAM (Low).

Bit 2 und 3 bestimmen im PC 128-Modus den Speichertyp, zwischen



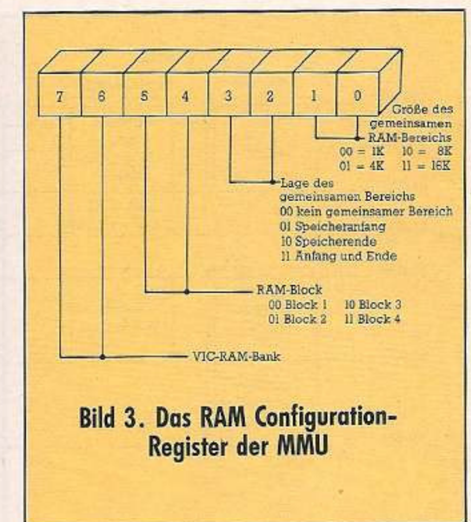



**Bild 3. Das RAM Configuration-Register der MMU**

den Adressen $8000 und $BFFF. Sind beide Bits »0«, wird auf Basic-High in der Memory Map, also dem zweiten Basic-ROM-Teil, zugegriffen. Ist Bit 2 »1«, wird ein internes ROM eingeblendet. In unserem Testgerät war noch ein Stecksockel auf der Platine für dieses ROM frei. Es ist jedoch geplant, dort einen deutschen Zeichensatz unterzubringen. Auch die Tastatur soll dann eine DIN-Belegung besitzen.

Interessant wird es erst richtig, wenn nur Bit 3 »1« ist, dann wird nämlich im Bereich von $8000 bis $BFFF ein Steckmodul eingeblendet. Sind beide Bits »1« sieht der PC 128 in diesem Bereich nur RAM.

Die nächsten beiden Bits, 4 und 5, haben die gleiche Funktion wie Bit 2 und 3, nur bestimmen sie den Speicheraufbau im Bereich von $C000 bis $FFFF. Zu bemerken ist, daß der Speicherbereich von $D000 bis $DFFF ein »ROM-Loch« darstellt. Man kann die Bits 4 und 5 setzen wie man will, der Computer entscheidet, ob I/O-Bereich oder das Zeichensatz-ROM in diesem Bereich eingeblendet wird.

Es ist also bei einem Steckmodul zu berücksichtigen, daß der Bereich von $D000 bis $DFFF tabu ist. Es können bis zu 32 KByte ROM eingeblendet werden, doppelt soviele wie beim C 64.

Bit 6 und 7 schließlich selektieren die RAM-Bank. Für die 128-KByte-Version des PC 128 ist nur Bit 6 wichtig. Ist es »0«, ist Bank 0 ausgewählt, ist es »1«, Bank 1.

**▲ Bild 5. Die neue Floppy-Station 1571 und der ebenfalls neue 1902-RGB/Composite-Monitor**



**▶ Bild 8. In der Version PC 128 sind Computer und 1571-Floppy im selben Gehäuse untergebracht**

## Verbindung nach draußen

Wie der C 64 hat der PC 128 einen Expansion-Port, der von beiden Betriebsmodi angesteuert wird (Bild 4). Im C 64-Modus entspricht die Pin-Belegung genau der des C 64.

### Speicherzugriff erlaubt

Im Gegensatz zum C 64 erlaubt der Expansion-Port des PC 128 einen direkten Speicherzugriff (DMA, Direct Memory Access). Direkter Speicherzugriff bedeutet, daß ohne Umwege über den Prozessor in den Speicher des PC 128 geschrieben oder der Speicher ausgelesen werden kann. Das wichtigste, um einen DMA realisieren zu können, ist, daß der Prozessor während des Zugriffs abgeschaltet bleibt. Beim PC 128 macht das der 8564-VIC. Er steuert den Daten- und Adreßbus so, daß Prozessor und DMA sich nicht ins Gehege kommen, was beim C 64 nicht immer sichergestellt ist. Bei einem gleichzeitigen Bus-Zugriff von Prozessor und externem Gerät erweist sich der Prozessor meist als der Schwächere, was zu ernsthaften Problemen führen kann.

Daß ein direkter Speicherzugriff ohne weiters machbar ist, eröffnet dem PC 128 gegenüber dem C 64 zusätzliche Einsatzgebiete in der Meßwerterfassung. Ein Meßgerät kann dadurch beispielsweise Meßwerte so schnell direkt in den Speicher schreiben, daß eine Echtzeiterfassung eines Meßvorganges möglich ist. Eine andere Möglichkeit wäre der Anschluß eines Festplatten-Laufwerkes. Die Daten könnten dann viel schneller in den RAM-Bereich geladen oder aus dem Arbeitsspeicher geholt werden, als wenn der Prozessor vorher jedes Bit ein paar mal »umdreht«.

### Der serielle Bus

Der serielle Bus und der Kassetten-Port des PC 128 sind von den Anschlüssen her identisch mit denen des C 64. Die Bedienung des seriellen Bus wurde überarbeitet, so daß der PC 128 zusammen mit dem neuen Commodore Laufwerk



**Bild 9. Der PC 128 mit RGB- und Composite-Monitor. Links das Programm und rechts die dadurch erzeugte Grafik.**

1571 wesentlich schneller speichern und laden kann als der C 64 (Bild 5).

> In der nächsten Ausgabe berichten wir über das neue Floppy-Disk-Laufwerk 1571 und den CP/M-Modus. Ferner erfahren Sie, was die deutsche Version des C 128 mehr bietet.

## GO 64 – Wie kompatibel ist der PC 128?

Kompatibilität war schon immer ein Reizwort für Commodore. Deswegen war Skepsis angesagt, ob der PC 128 wirklich kompatibel zum C 64 ist.

Also haben wir eine Zahl von Programmen ausprobiert, die direkt oder indirekt über einen Kopierschutz im Betriebssystem herumpfuschen oder sonstige Gemeinheiten anstellen, die jeden Nicht-C 64 sofort zum Aussteigen bewegen würden. Erster Testkandidat war Hypra-Load. Einige Probeläufe zeigten, daß sich hier in Verbindung mit der 1541 überhaupt keine Probleme ergeben. Damit dürfte gesichert sein, daß alle Programme mit geänderten Busroutinen einwandfrei funktionieren.

Ein Blick ins Innere des PC 128 (Bild 6 und 10) zeigte auch, daß immer noch dieselben Bus-Bausteine verwendet werden. Gleichzeitig entspricht die Taktfrequenz des PC 128 im C 64-Modus der des C 64, so daß hier eigentlich auch keine Probleme erwartet wurden. Nächstes Testobjekt war ein Kopierprogramm, das intensiven Gebrauch von illegalen Opcodes macht. Mit Opcodes bezeichnet man den Befehlssatz des Prozessors. Illegale Opcodes sind Befehle, die der Hersteller des Prozessors eigentlich gar nicht vorgesehen hat. In Wirklichkeit bewirken aber manche von ihnen auch beim C 64 schon etwas. Und einige Programme nutzen sie. Es hätte also sein können, daß der 8502-Prozessor einige dieser beim 6502 an sich undefinierten Opcodes benutzt. Doch traten hier keine Probleme auf. Auch alle anderen kopiergeschützten (und nicht kopiergeschützten) Diskettenprogramme konnten wir ohne Schwierigkeiten laden und benutzen.

Wir verwendeten bei unserem Test das bekannte 1541-Floppy-Laufwerk (Bild 13). Wie es sich mit dem Nachfolgemodell, der 1571, verhält, berichten wir im zweiten Teil unseres PC 128-Tests.

Getestet wurden von uns diverse Spiele wie Ghostbusters und Pit Stop II. Auch hier ein eindeutiges Ergebnis: Grafik und Musik stimmen mit dem C 64 überein. Da auch die Datasette an den PC 128 angeschlossen werden kann, standen als nächstes Kassettenprogramme auf dem Plan: Diverse Spiele, zum Teil mit Turboladern und Autostart versehen, liefen genauso problemlos wie die Diskettenprogramme.

Letzter, und unserer Ansicht nach härtester Prüfstand: Module im Expansionport. Auch hier, wie fast schon erwartet, keine Probleme, egal ob Soccer oder GBasic. Insbesondere das GBasic-Modul, das ja im Modul selbst noch eine Bank-Switching-Elektronik enthält, also zwischen zwei Speicherbausteinen hin und herschaltet, wie der PC 128 selbst, dürfte der letzte Beweis dafür sein, daß der PC 128 im C 64-Modus vollkommen software-kompatibel zum C 64 ist. Wie es mit diversen Hardwareerweiterungen, beispielsweise Turbo Access oder Speeddos aussieht, ist noch ungewiß. Sicher ist, daß hier zumindest die Platinen dieser Erweiterungen geändert werden müssen, da beim PC 128 das Kernal und das Basic in einem einzigen 16 KByte ROM und nicht, wie beim C 64, in 2 ROMs zu je 8 KByte enthalten ist.

Klares Fazit unseres Kompatibilitätstests: Wer schon einen C 64 mit Floppy 1541 hat, der kann zumindest seinen Computer beruhigt weiterverkaufen, steigt er auf den PC 128 um. Es gibt keinen Programmtyp, den wir nicht getestet haben; wir versuchten es mit Programmen, die auf übelste Art im Betriebssystem herumspringen: keine Chance. Der PC 128 ist voll kompatibel zum C 64, selbst wieder bei den Joystickanschlüssen (Bild 7).

## Super-Basic 7.0

Beim Basic-Interpreter zeigt sich der PC 128 und der PC 128D (Bild 8) ohne Zweifel von einer seiner stärksten Seite: Das Basic 7.0 enthält alle Befehle und Funktionen der Basic-Versionen 2.0 (C 64), 3.5 (C 16 und Plus/4) und 4.0 (CBM 80xx). Damit stehen bereits leistungsfähige Grafikbefehle wie DRAW, BOX oder CIRCLE sowie viele Diskettenkommandos zur Verfügung. Doch damit nicht genug. Zusätzlich enthält das 7.0-Basic eine Reihe spezieller Befehle zur Steuerung von Sprites und zur einfachen Programmierung des Synthesizer-Bausteins (SID). Die zusätzlich zum 2.0-Basic vorhandenen Befehle und Funktionen sind in Tabelle 1 beschrieben.

Schon eine erste, oberflächliche Betrachtung dieser Tabelle läßt eine neue Dimension der Basic-Pro-

grammierung erahnen. Endlose DATA-Orgien und wüster GOTO-Dschungel gehören mit diesem Basic endgültig der Vergangenheit an.

Formatierte Zahlenausgabe mittels PRINT USING ist dabei ebenso selbstverständlich wie Befehle zur Abfrage von Joystick, Lightpen und Paddles.

Mit WINDOW läßt sich ein Bildschirmfenster definieren, auf das sich anschließend alle PRINT- und INPUT-Befehle beziehen. Der Befehl mit dem beziehungsreichen Namen SLEEP läßt den PC 128 denn auch tatsächlich für die angegebene Zeit schlafen: »SLEEP 5« hält das Programm fünf Sekunden lang an. So spart man sich das umständliche Hantieren mit leeren FOR...NEXT-Schleifen für oftmals sinnvolle Verzögerungen im Programmablauf. Zeiten zwischen einer Sekunde und 18 Stunden (!) sind programmierbar, womit sich die Frage aufwirft, wer seinen Computer während eines Programms wohl für mehr als eine Minute anhalten will.

Zwei weitere ungewöhnliche Befehle fallen sofort auf, nämlich SLOW und FAST. Mit diesen Befehlen kann der PC 128 zwischen 1 MHz Taktfrequenz (SLOW) und 2 MHz umgeschaltet werden. Nach dem Einschalten läuft der Computer mit einem Takt von 1 MHz, also mit ähnlicher Geschwindigkeit wie der C 64. Durch die komplizierte Art der Speicherverwaltung mit den verschiedenen Speicherbänken für Programme, Variablen und Betriebssystem/Basic ist das PC 128-Basic prinzipiell geringfügig langsamer als das C 64-Basic, dies wird jedoch, wie schon erwähnt, einerseits durch den wesentlich leistungsfähigeren Befehlssatz mehr als aufgewogen, zum anderen kann durch den FAST-Befehl die Abarbeitungsgeschwindigkeit exakt verdoppelt werden.

So schön das im Prinzip auch ist, die Geschwindigkeitsvorteile des FAST-Modus muß man sich mit dem bereits erwähnten Nachteil erkaufen.

Doch wenden wir uns wieder dem Basic selbst und damit erfreulicheren Dingen zu.

Eine Reihe von Befehlen dient ausschließlich der bequemeren Programmentwicklung: AUTO gibt bei der Programmeingabe automatisch die Zeilennummern vor, mit TRON kann in der Testphase eines Programms eine Trace-Funktion eingeschaltet werden. Es werden dann auf dem Bildschirm die Zeilennummern der gerade abgearbeiteten Basic-Zeilen angezeigt. Dies bewährt sich insbesondere bei Fehlern in der Programmlogik. Eine falsch gesetzte IF-Abfrage wird damit zum Beispiel schnell erkannt — man sieht ja, wohin das Programm springt. Zu Testzwecken kann TRON natürlich auch im Programm verwendet werden. Am Anfang eines »verdächtigen« Programmteils fügt man einfach den TRON-Befehl ein, am Ende dieses Abschnittes wird die Trace-Funktion mit TROFF wieder außer Betrieb gesetzt.

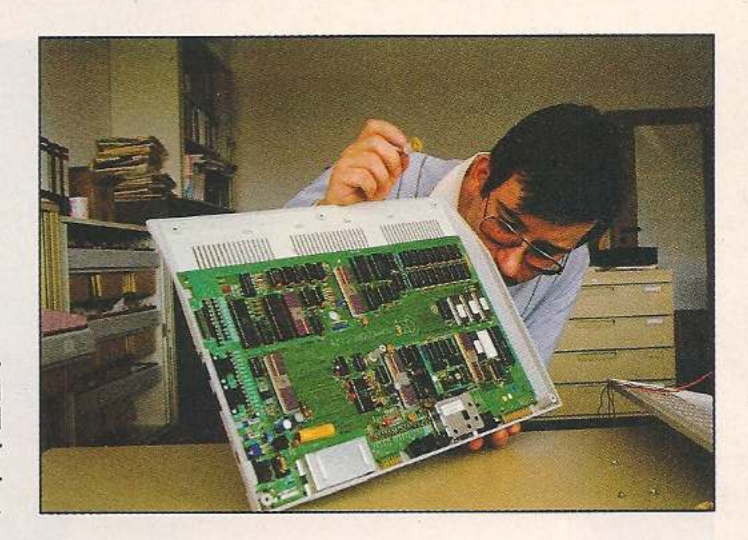


**Bild 10. Hardware-Spezialist und Redakteur Harald Meyer bei der Analyse der PC 128-Platine.**

Der RENUMBER-Befehl dient zum Umnumerieren des gesamten Programms oder auch nur einzelner Teile davon. Während jedoch RENUMBER beim bekannten Simons-Basic für den C 64 weder GOTO- noch GOSUB-Adressen ändert (und mithin eher ein Problem als ein Hilfsmittel darstellt), korrigiert das 7.0-Basic automatisch alle Zeilennummern hinter GOTO, GOSUB, THEN, ELSE, RESTORE und RESUME und sogar bei Abfragen von Fehlerzeilen mittels der Spezialvariablen EL in einer Fehlerbehandlungsroutine. Wobei wir gleich bei einem weiteren interessanten Aspekt des 7.0-Basic wären.

## Fehlerbehandlung ohne Programmabbruch

Während der C 64 bei jedem auftretenden Fehler unerbittlich sein Programm mit einer entsprechenden Meldung beendet, bietet der PC 128 hier einiges mehr an Flexibilität. Mit der TRAP-Anweisung können alle auftretenden Fehler während des Programmlaufes abgefangen werden. Zum Beispiel wird nach der Anweisung »TRAP 500« beim Auftreten eines Fehlers das Programm nicht unterbrochen, sondern es wird in eine Fehlerbehandlungsroutine (hier ab Zeile 500) verzweigt. Alle wichtigen Daten über den Fehler werden in Systemvariablen gespeichert und können von der (vom Programmierer zu schreibenden) Basic-Routine ab Zeile 500 ausgewertet werden: EL enthält die Zeilennummer, in der der Fehler auftrat, ER enthält die Fehlernummer und ERR$ liefert die Fehlermeldung im Klartext. Die Fehlerbehandlungsroutine kann diese Variablen auswerten, um gezielt Maßnahmen zu ergreifen. Anschließend sollte das Programm natürlich weiter fortgesetzt werden können. Dazu dient die RESUME-Anweisung, die eine Fehlerbehandlung abschließt (vergleichbar mit RETURN bei Unterprogrammen). RESUME kann auf drei verschiedene Arten verwendet werden. RESUME ohne weitere Parameter kehrt zu der Anweisung zurück, die den Fehler verursacht hat und setzt das Programm dort ganz normal fort. In diesem Falle muß natürlich in der Fehlerbehandlungsroutine die Fehlerursache behoben worden sein, sonst tritt der Fehler sofort wieder auf. Ein gutes Beispiel ist der Test, ob der Drucker eingeschaltet ist:

```
10 TRAP 90 : OPEN 1,4
20 PRINT#1,"Drucker OK"
30 END
90 IF ER=5 AND EL=10 THEN
PRINT "Bitte Drucker einschalten
und Taste drücken" : GETKEY A$
95 RESUME
```

Dieses kleine Demo-Programm gibt den Text »Drucker OK« auf einem angeschlossenen Drucker aus. Falls der Drucker nicht eingeschaltet sein sollte, würde der OPEN-Befehl in Zeile 10 normalerweise zur Fehlermeldung »Device not present« führen. Diese Meldung wird aber durch den TRAP-Befehl im Falle eines Falles abgefangen und statt dessen zur Zeile 90 verzweigt, wo

nach Überprüfung auf Fehlernummer und -zeile der Benutzer höflich aufgefordert wird, doch bitteschön den Drucker einzuschalten. Der Befehl GETKEY wartet anschließend auf einen Tastendruck, worauf das Programm durch den RESUME-Befehl wieder zum OPEN-Kommando zurückkehrt.

Soll das zum Fehler führende Kommando nicht nochmals ausgeführt werden, dann muß die Fehlerbehandlungsroutine mit RESUME NEXT abgeschlossen werden, wodurch mit dem nächsten Befehl nach der Fehlerursache weitergemacht wird. In besonderen Fällen kann es nach einem Fehler nützlich sein, ganz woanders im Programm fortzufahren. In einem solchen Falle kann hinter RESUME eine Zeilennummer angegeben werden, an der das Programm fortgesetzt werden soll.

Mit diesen Möglichkeiten zur Fehlerbehandlung im Programm selbst steht dem Programmierer ein leistungsfähiges Werkzeug zur Verfügung. Und sollte in der Entwicklungsphase eines Programms doch einmal ein Fehler auftreten, dann genügt ein Druck auf die HELP-Taste, um die fehlerhafte Zeile aufzulisten. Der Teil der Zeile, der den Fehler verursachte, wird dabei revers dargestellt.

Natürlich lassen sich auch von der Diskettenstation gemeldete Fehler in ähnlich eleganter Weise abfragen. Statt umständlich die Zeile

```
1 OPEN 1,8,15 : INPUT#1,A,B$,C,D :
PRINT A,B$,C,D : CLOSE 1 : END
```

einzugeben (und dabei womöglich sein Programm zu überschreiben) tippt man beim 7.0-Basic einfach »?DS$« und erhält die gleiche Meldung. Die Systemvariable DS$ enthält nämlich den Fehlerstatus der Diskettenstation als Klartext, die Systemvariable DS den entsprechenden Fehlercode.

Überhaupt stehen beim PC 128 alle Diskettenbefehle als Basic-Kommandos zur Verfügung. SCRATCH beispielsweise löscht ein File von der Diskette, DIRECTORY oder CATALOG listen das Inhaltsverzeichnis ohne Programmverlust, mit DLOAD, DSAVE und DVERIFY spart man sich das lästige »,8«. BLOAD und BSAVE dienen zum Laden/Abspeichern beliebiger Speicherinhalte (Maschinenprogramme, Grafik etc.). Neu sind auch eine Reihe von Befehlen zur komfortablen Verwaltung sequentieller und relativer Dateien. Mit RECORD kann beispielsweise direkt auf einen Datensatz einer relativen Datei zugegriffen werden, APPEND ermöglicht das Anfügen weiterer Datensätze bei sequentiellen Dateien.

Das 7.0-Basic bietet eine ganze Reihe spezieller Schleifen- und Strukturbefehle zur GOTO-freien, strukturierten Programmierung. Da wäre zunächst einmal die Erweiterung der IF..THEN — Abfrage um die ELSE-Klausel. Bisher mußte man beispielsweise alternative Entscheidungen wie folgt programmieren:

## Programmieren ohne GOTO

```
10 IF A$="N" THEN PRINT
"NEIN" : GOTO 30
20 PRINT "JA"
30 REM Hier geht's weiter
```

Im 7.0-Basic reicht dazu eine Zeile, und die ist noch um einiges leichter verständlich:

```
10 IF A$="N" THEN PRINT
"NEIN" : ELSE PRINT "JA"
```

Wenn A$ gleich »N« ist, dann wird »nein« gedruckt, sonst »ja«.

Leider ist die ELSE-Anweisung in dieser Form auf eine Zeile beschränkt. Abhilfe schafft hier die Klammerung mit BEGIN...BEND.

Alle zwischen BEGIN und BEND stehenden Basic-Zeilen stellen einen Block dar, der vom Basic-Interpreter genauso wie eine einzelne Zeile behandelt wird. Deshalb wird BEGIN...BEND besonders vorteilhaft bei IF-Abfragen benutzt:

```
10 INPUT "HEISST DEIN COMPU-
TER COMMODORE ODER
SCHNEIDER ?";C$
20 IF C$="COMMODORE" THEN
BEGIN
30 :     PRINT "PC 128 KAUFEN !"
40 : BEND : ELSE BEGIN
50 :     PRINT "VERRAETER !"
60 BEND
```

Man beachte, daß sich die IF-Anweisung insgesamt von Zeile 20 bis Zeile 60 erstreckt. In diesem Beispiel erhält man den Ratschlag, sich einen PC 128 zu kaufen, falls der Computer »Commodore« heißt. Hat man jedoch »Schneider« (oder etwas anderes) als Namen angegeben, wird man sofort als »Verräter« tituliert.

Natürlich können derartige IF...THEN...ELSE-Abfragen mit BEGIN...BEND auch geschachtelt werden, das heißt, man kann sowohl in den THEN- als auch in den ELSE-Teil weitere IF-Abfragen einbauen.

Somit lassen sich auch größere Programmblöcke ohne GOTO programmieren. Der Verzicht auf GOTO erhöht nicht nur die Übersichtlichkeit, sondern auch die Geschwindigkeit beim Programmlauf. Bei jedem GOTO-Befehl muß der Basic-Interpreter nämlich erstens die Zeilennummer, die im Programm ja als Dezimalzahl steht, in das interne binäre Format umrechnen und zweitens dann auch noch die angegebene Zeile suchen. Ein weiterer Vorteil: In den Programmbefehlen selbst kommen keine weiteren Zeilennummern mehr vor, das Beispielprogramm kann unverändert in allen möglichen Zeilenbereichen laufen.

Aber nicht nur Verzweigungen lassen sich derart elegant programmieren, besonders bei Schleifen, also bei Wiederholungen von bestimmten Programmteilen, spielt das 7.0-Basic seine Stärken erst richtig aus. Es ist ja vom C 64 her bekannt, daß eine FOR...NEXT-Schleife um einiges schneller ist als die gleiche Schleife mittels IF und GOTO programmiert. Nachteilig bei der FOR...NEXT-Schleife ist, daß die Anzahl der Schleifendurchläufe schon bei Eintritt in die Schleife bekannt sein muß. Dieser Nachteil wird durch die neue, schnelle DO...LOOP-Schleifenstruktur behoben. Wie FOR...NEXT umklammert auch DO...LOOP einen beliebig großen Programmteil. Die Wirkung des DO-Befehls besteht einfach darin, daß der Basic-Interpreter sich den Anfang der Schleife »merkt«. Bei Erreichen des zugehörigen LOOP wird dann sehr schnell, ohne Suchzeiten, zum DO zurückgesprungen. Es ergibt sich also eine »unendliche Schleife« zwischen DO und LOOP. Um diese Schleife dennoch verlassen zu können, ist der EXIT-Befehl vorgesehen. Die Wirkung von EXIT besteht einfach darin, die Programmausführung hinter LOOP ganz normal fortzusetzen. Normalerweise wird EXIT daher von einer Bedingung abhängig gemacht. Beispiel:

```
10 X=1
20 DO
30 : X=X*2 : PRINT X
40 : IF X>1500 THEN EXIT
50 LOOP
```

Der Wert X wird hier solange verdoppelt und ausgedruckt, bis der Wert 1500 überschritten wird.

Neben dieser unbedingten DO...LOOP-Schleife sind noch zwei von Bedingungen abhängige Formen vorgesehen. DO WHILE ... LOOP wird so lange ausgeführt, wie eine nach WHILE stehende Bedingung wahr ist:

```
10 DO WHILE A$="" : GET A$ :
LOOP
```

Solange keine Taste gedrückt wird, ist A$ immer leer, die WHILE-

Bedingung also erfüllt. Die Schleife wird daher erst verlassen, wenn eine Taste gedrückt wird.

Die DO UNTIL-Schleife wird dagegen nicht ausgeführt, solange die Bedingung wahr ist, sondern im Gegenteil so lange, bis die hinter UNTIL angegebene Bedingung wahr wird.

Natürlich können auch bei DO WHILE oder DO UNTIL zusätzliche EXITs in die Schleife eingebaut werden, was die Leistungsfähigkeit dieser Anweisungen noch erhöht.

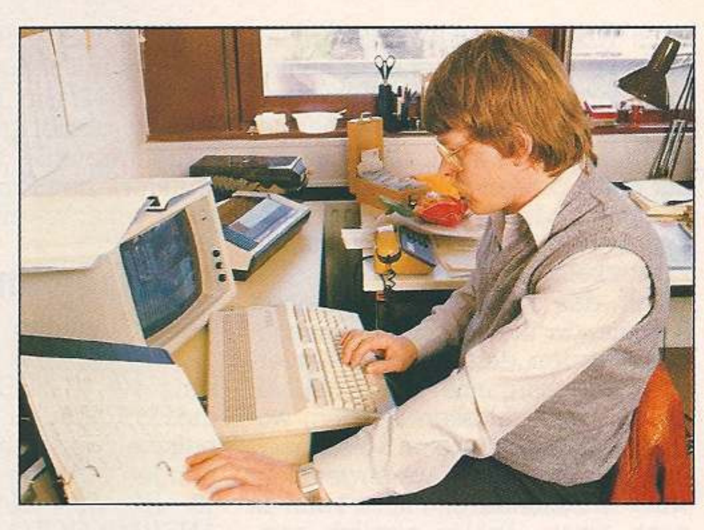


**Bild 11. In unermüdlicher Kleinarbeit wühlt sich Georg Klinge durch Handbuch und Speicherorganisation.**

## Die Grafik ist für alle da

Um hochauflösende Grafik auf dem C 64 zu realisieren gibt es außer dem Kauf diverser Basic-Erweiterungen (oder dem Abtippen von 64'er-Listings) im wesentlichen nur die Alternative, selbst zum Maschinensprache-Profi zu werden — ungefähr so, als wenn man Radio- und Fernsehmechaniker werden müßte, um an seinem Farbfernseher die Farbe einstellen zu können. Ein sicherlich unhaltbarer Zustand, dessen Änderung Commodore allerdings bereits mit dem 3.5-Basic des C 16 in Angriff genommen hatte. Die hochauflösende Grafik des PC 128 ist genauso wie die des C 64/C 16 aufgebaut. Insgesamt 64000 Einzelpunkte können getrennt angesprochen werden, was einer Auflösung von 320 x 200 Punkten entspricht. Daneben ist ein Mehrfarbenmodus mit einer Auflösung von 160 x 200 Punkten vorgesehen, bei dem jeder Einzelpunkt eine von vier Farben haben kann.

Der große Unterschied zum C 64 liegt darin, daß die PC 128-Grafik voll vom Basic unterstützt wird. Befehle wie DRAW, BOX oder CIRCLE ermöglichen schnelles und unkompliziertes Zeichnen geometrischer Figuren von Linien über Drei-, Vier- und Mehrecke bis hin zu Kreisen und Ellipsen. Alle Figuren können beliebig vergrößert, verkleinert und sogar gedreht oder ausschnittsweise dargestellt werden. PAINT füllt geschlossene Flächen aus, SCALE dient zur Skalierung der Zeichenfläche und SCNCLR löscht den Grafikbildschirm.

Alle Grafikbefehle arbeiten sowohl im Hochauflösungs- wie auch im Mehrfarben-Modus. Mit dem Befehl GRAPHIC wird der gewünschte Grafik-Modus eingestellt. Zur Wahl stehen Text mit 40 Zeichen Text, Hochauflösung, Hochauflösung mit Textfenster, Mehrfarbengrafik, Mehrfarbengrafik mit Textfenster und schließlich Text mit 80 Zeichen pro Zeile.

Commodore spricht zwar von einer Auflösung von 640 x 200 Punkten, die im 80-Zeichen-Modus möglich ist, an wirklich doppelt auflösende Grafik ist dabei aber nicht zu denken: Die 640 Punkte ergeben sich als reines Rechenexempel aus 80 x 8, also 80 Zeichen mal 8 Punkte je Zeichen (Zeichenmatrix 8 x 8). Die uns beim Test vorliegende Vorab-Version des Handbuchs schweigt sich völlig über eine doppeltauflösende Grafik aus, ebenso das Hardware-Manual. Versuche ergaben, daß der GRAPHIC-Befehl tatsächlich nur mit den genannten sechs Parametern (40-Zeichen-Text, Hochauflösung, Hochauflösung mit Textfenster, Mehrfarben, Mehrfarben mit Textfenster, 80-Zeichen-Text) funktioniert, alles andere ergibt einen »Illegal Quantity Error«. Um es ganz deutlich zu sagen: Wirkliche Grafik mit einer Auflösung von 640 x 200 Punkten ist nach unseren bisherigen Erfahrungen mit dem PC 128 zumindest ohne ausgiebiges Tricksen nicht möglich. Insbesondere beziehen sich alle Grafikbefehle des 7.0-Basic ausschließlich auf die vom C 64 her bekannte 320 x 200 Punkte-Auflösung (und natürlich wahlweise auf den Mehrfarbenmodus mit 160 x 200 Punkten).

Ein weiterer Wermutstropfen: Die ganze schöne Grafik, Sprites und 40-Zeichen-Text sind ausschließlich über einen Composite-Monitor verfügbar, auf einem RGB-Monitor tut sich überhaupt nichts. Andersherum ist die 80-Zeichen-Textdarstellung nur über RGB (oder natürlich einen monochromen Monitor) möglich.

Der verblüffte Anwender stellt spätestens jetzt fest, daß er einfach einen Monitor zu wenig hat. Damit dürfte Commodore sich die Urheberrechte am ersten Zwei-Monitor-Heimcomputer der Welt gesichert haben. Wohlgemerkt, man hat nicht die Wahl zwischen Composite und RGB, sondern braucht unbedingt einen Composite-Monitor für 40-Zeichen, Grafik und Sprites und ebenso unbedingt entweder einen RGB- oder einen SW-Monitor für 80 Zeichen (Bild 9 und 12). Abhilfe schafft hier der neue 1902-Monitor von Commodore, der speziell zum PC 128 entwickelt wurde und sowohl über einen Composite- als auch über einen RGB-Eingang verfügt. Zwischen beiden Betriebsarten des Monitors wird mit einem kleinen Schalter an der Frontseite hin- und hergeschaltet — eine softwaremäßige Umschaltung ist nicht vorgesehen. Man kann daher nur wünschen, daß der Umschalter stabil genug gebaut ist — er wird oft betätigt werden müssen.

Als Fazit zur PC 128-Grafik bleibt festzuhalten, daß sie von der Auflösung her dem durch den C 64 gesetzten Standard (320 x 200 Punkte) entspricht und wie beim C 16 vorbildlich durch das Basic unterstützt wird.

## Shapes, Sprites und Sprite-Editor

Wenn von Grafik die Rede ist, dürfen natürlich Shapes und Sprites nicht fehlen. Hinsichtlich dieser beweglichen Grafikobjekte ist beim PC 128 eine gelungene Synthese von C 64-Hardware und C 16-Software zu verzeichnen. Vom C 64 stammen die acht Sprites, freiprogrammierbare, bewegliche Grafikobjekte, die von der Hardware (VIC-Chip) erzeugt und in den Bildschirm eingeblendet werden. Sprites können sowohl im 40-Zeichen-Textmodus als auch in den verschiedenen Grafik-Modi erzeugt werden, nicht allerdings im 80-Zeichen-Modus, denn der VIC, der sie erzeugt, ist nicht RGB-fähig.



Aus dem 3.5-Basic des C 16 wurde das Konzept der softwaremäßig erzeugten Shapes übernommen. Shapes sind rechteckige Ausschnitte aus der hochauflösenden oder der Mehrfarben-Grafik, die in Stringvariablen abgespeichert werden und daraus auch wieder auf den Bildschirm gebracht werden können. Da es sich um reine Grafikelemente handelt, können sie weder im 40- noch im 80-Zeichen-, sondern nur im Grafik-Modus dargestellt werden. Mit »SSHAPE X$, 100, 100, 150, 120« wird beispielsweise der Inhalt des Rechtecks mit linker oberer Ecke (100, 100) und rechter unterer Ecke (150, 120) aus der hochauflösenden Grafik in der Stringvariablen A$ abgelegt. Mit »GSHAPE A$, X, Y« wird die in A$ enthaltene Grafik-Information an der Grafikposition X, Y wieder auf den Bildschirm gebracht. Neben den Sprites sind die Shapes also eine zweite, leistungsfähige Möglichkeit zur Darstellung grafischer Objekte und eröffnen in Zusammenhang mit der hohen Speicherkapazität des PC 128 völlig neue Möglichkeiten für Spiele in hochauflösender Grafik.

## Integrierter Sprite-Editor

Das Basic 7.0 enthält sogar einen integrierten Sprite-Editor, mit dem man direkt am Bildschirm das Punktmuster des gewünschten Sprites entwerfen kann. Mit dem SPRITE-Befehl werden für jedes Sprite folgende Attribute gesetzt: Aktivität, Farbe, Priorität, Dehnung in X- und Y-Richtung und Modus (hochauflösend oder Mehrfarben).

Mit »SPRITE 4,1,6,1,1,0,0« wird zum Beispiel das Sprite Nr. 4 aktiviert (1). Es wird in der Farbe Grün (6) angezeigt, hat Priorität über bereits angezeigte Bildschirmdaten (1), ist in X-Richtung gedehnt (1), in Y-Richtung nicht gedehnt (0) und wird im Hochauflösungs-Modus angezeigt (0).

Um umgekehrt die Attribute eines bereits definierten Sprites zu bestimmen, kann die RSPRITE-Funktion verwendet werden.

Mit dem SPRSAV-Kommando können die Daten eines Sprites in einer Stringvariablen abgelegt werden oder umgekehrt aus einem String ausgelesen werden.

Die Steuerung der Sprites erfolgt über den MOVESPR-Befehle mit dem ein Sprite an eine bestimmte Bildschirmposition gesetzt werden kann. Die Positionsangabe kann entweder in absoluten Koordinaten oder auch relativ zur bisherigen Position erfolgen. Doch damit noch

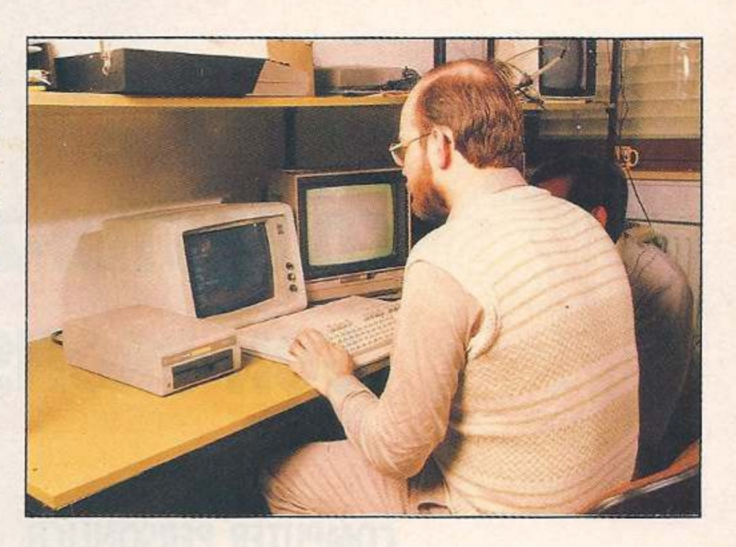


**Bild 12. Das gesamte 7.0-Basic wurde von Volker Everts, dem Sprachenspezialisten, konsequent durchprobiert und ausgetestet.**

### Diese Befehle sind im

| Befehl | Beschreibung |
|---|---|
| AUTO | Automatische Zeilennumerierung |
| APPEND | Öffnet eine sequentielle Datei zum Datenanfügen |
| BACKUP | Kopiert eine komplette Diskette |
| BANK | Wählt Speicherbank für PEEK, POKE und SYS |
| BEGIN...BEND | Faßt mehrere Basic-Zeilen zu einem Block zusammen |
| BOOT | Lädt und startet CP/M von Diskette |
| BOX | Zeichnet Rechtecke |
| BSAVE | Speichert beliebige Speicherbereiche auf Floppy |
| BUMP | Liefert bei Sprite-Kollisionen die Sprite-Nummer |
| CATALOG | Listet Inhaltsverzeichnis der Diskette |
| CHAR | Fügt Text in die hochauflösende Grafik ein |
| CIRCLE | Zeichnet Kreise, Ellipsen und Vielecke |
| COLLECT | Löscht offene Dateien und reorganisiert Diskette |
| COLLISION | Dient zur Sprite-Kollisions-Abfrage |
| COLOR | Setzt Farben für Text und Grafik |
| CONCAT | Verbindet zwei sequentielle Dateien miteinander |
| COPY | Kopiert eine Disketten-Datei |
| DCLEAR | Schließt alle Kanäle zur Diskettenstation |
| DCLOSE | Schließt Kanal zur Diskettenstation |
| DEC | Dezimalwert einer Hexadezimalzahl |
| DELETE | Löscht einen Zeilenbereich aus dem Programm |
| DIRECTORY | Disketteninhaltsverzeichnis (wie CATALOG) |
| DLOAD | Lädt ein Programm von Diskette |
| DOPEN | Öffnet Kanal zur Diskettenstation |
| DO...LOOP | Programmschleife. LOOP springt immer zu DO zurück. |
| DRAW | Setzt Punkte und zeichnet Linien |
| DSAVE | Speichert ein Programm auf Diskette |
| DS | Ergibt den Fehlerstatus des Diskettenlaufwerks |
| DS$ | Enthält Fehlerstatus der Floppy im Klartext |
| DVERIFY | Überprüft Programmspeicherung auf Disk |
| EL | Enthält Zeilennummer bei Auftreten eines Fehlers |
| ELSE | Alternative bei IF-THEN, falls Bedingung nicht erfüllt |
| ENVELOPE | Definiert Hüllkurve für Synthesizer |
| ER | Liefert den Code des zuletzt aufgetretenen Fehlers |
| ERR$ | Liefert Fehlermeldung im Klartext |
| EXIT | Dient zum Verlassen einer DO...LOOP-Schleife |
| FAST | Schaltet auf doppelte Geschwindigkeit (2 MHz Takt) |
| FETCH | Holt Daten aus beliebiger Speicherbank (RAM-Floppy) |
| FILTER | Setzt die Klangfilter-Parameter für den SID |
| GETKEY | Wartet auf Tastendruck |
| GO64 | Schaltet in den C 64-Modus |
| GRAPHIC | Wählt Grafik-Modus aus |
| GSHAPE | Schreibt ein Shape aus einem String auf den Bildschirm |
| HEADER | Dient zum Formatieren von Disketten |
| HELP | Listet nach Fehlermeldung die Fehlerzeile am Bildschirm |
| HEX$ | Wandelt Dezimalzahlen in Hexadezimal-Strings |
| INSTR | Ergibt Position eines Teilstrings in einem anderen String |
| JOY | Fragt Joystickposition ab |
| KEY | Dient zur Belegung der Funktionstasten |

nicht genug. Gibt man zusätzlich noch eine Geschwindigkeit als Zahlenwert zwischen 1 und 15 an, so gleitet das Sprite automatisch an die angegebene neue Position. Durch Setzen von Plus- oder Minuszeichen vor die Koordinatenangaben werden aus den absoluten Koordinaten relative Koordinaten. Ohne Geschwindigkeitsangabe erscheint das Sprite sofort an der neuen Position. »MOVESPR 7,-30,+40« versetzt Sprite 7 augenblicklich um 30 Punkte nach links und um 40 Punkte nach oben. Beim C 64 kann man durch PEEKen in die Sprite-Kollisionsregister des VIC feststellen, ob ein Sprite mit einem anderen Sprite oder mit Hintergrunddaten kollidiert ist. Beim PC 128 bedient man sich für den gleichen Zweck um einiges eleganter der COLLISION-Anweisung. Damit kann eine automatische Programmunterbrechung bei Eintritt entweder einer Sprite/Sprite- oder einer Sprite/Hintergrund-Kollision programmiert werden. »COLLISION 1,500« hat beispielsweise folgende Bedeutung: Falls im weiteren Verlauf des Programms eine Sprite-Sprite-Kollision (Kennziffer 1) auftritt, dann wird das laufende Basic-Programm unterbrochen, und es wird ein Unterprogramm ab Zeile 500 ausgeführt. Nach dem RETURN wird das Programm an der Unterbrechungsstelle fortgesetzt.

COLLISION und MOVESPR sind leistungstarke Befehle, die ein Basic-Programm hinsichtlich der Sprite-Steuerung sehr stark entlasten. Um ein Sprite quer über den Bildschirm zu bewegen, muß man beim C 64 noch mit einer FOR... NEXT-Schleife arbeiten; um Kollisionen festzustellen, war daneben noch ein ständiges PEEKen in die Kollisionsregister des VIC nötig. Beim PC 128 reichen zwei Basic-Befehle, die zudem noch interruptgesteuert arbeiten, so daß das Basic-Programm während der Bewegung der Sprites weiterlaufen kann, bei einer eventuell auftretenden Sprite-Kollision dagegen automatisch unterbrochen wird, um schnell darauf reagieren zu können.

## Basic 7.0 dazugekommen

| | |
|---|---|
| LOCATE | Positioniert den Grafik-Cursor |
| MID$ | Ermöglicht jetzt auch Wertzuweisung an Teilstrings |
| MONITOR | Ruft den eingebauten Maschinensprache-Monitor auf |
| MOVESPR | Bewegt ein Sprite über den Bildschirm |
| PAINT | Füllt einen Bereich der hochauflösenden Grafik aus |
| PEN | Fragt Lightpen ab |
| PLAY | Spielt die in einem String abgelegte Tonfolge |
| POINTER | Ergibt die Adresse einer Variablen im Speicher |
| POT | Fragt Paddles ab |
| PRINT USING | Erlaubt formatierte Zahlenausgabe |
| PUDEF | Definiert Steuerzeichen für PRINT USING |
| RCLR | Liefert gewählten Farbcode für Text und Grafik |
| RECORD | Positioniert Schreib-/Lesezeiger bei relativen Dateien |
| RENAME | Dient zum Umbenennen von Diskettendateien |
| RENUMBER | Numeriert das Basic-Programm neu |
| RESTORE | Setzt DATA-Zeiger auf beliebige Zeilennummer |
| RESUME | Rückkehr aus einer Fehlerbehandlungsroutine |
| RGR | Liefert die Nummer des eingestellten Grafik-Modus |
| RREG | Weist Variablen die Werte der Prozessorregister zu |
| RSPRCOLOR | Liefert den aktuellen Code des Mehrfarbenmodus für Sprites |
| RSPPOS | Liefert Position und Geschwindigkeit eines Sprites |
| RSPRITE | Ergibt je nach Parameter alle Sprite-Attribute |
| RWINDOW | Liefert Parameter des eingestellten Bildschirmfensters |
| SCALE | Ermöglicht Maßstabswahl bei hochauflösender Grafik |
| SCNCLR | Löscht Text- oder Grafikbildschirm |
| SCRATCH | Löscht eine Diskettendatei |
| SSHAPE | Speichert ein Shape in eine Stringvariable |
| SLEEP | Hält die Programmausführung für eine wählbare Zeit an |
| SLOW | Schaltet von 2 MHz auf 1 MHz Takt zurück |
| SOUND | Erzeugt Toneffekte mit wählbarer Frequenz und Dauer |
| SPRCOLOR | Setzt Mehrfarben-Modus-Farben für Sprites |
| SPRDEF | Ruft den integrierten Sprite-Editor auf |
| SPRITE | Setzt Sprite-Atribute |
| SPRSAV | Speichert ein Sprite in einem String oder umgekehrt |
| STASH | Überträgt Daten in eine Speicherbank (RAM-Floppy) |
| SWAP | Tauscht Daten zwischen zwei Speicherbänken aus |
| TEMPO | Setzt Abspieltempo für PLAY-Anweisung |
| TRAP | Verzweigt im Fehlerfall zu einer Fehlerbehandlungsroutine |
| TROFF | Schaltet Programmablaufverfolgung (Trace) aus |
| TRON | Schaltet Trace ein |
| UNTIL | Setzt Bedingung für DO...LOOP fest (DO UNTIL ...) |
| VOL | Setzt Lautstärke für die SOUND-Anweisung |
| WHILE | Setzt Bedingung für DO...LOOP fest (DO WHILE ...) |
| WIDTH | Setzt die Strichstärke für alle Grafikbefehle |
| WINDOW | Definiert ein Bildschirmfenster |
| XOR | Liefert die Exklusiv-Oder-Verknüpfung zweier Werte |

**Tabelle 1. Die Befehle von Basic 2.0 (C 64/VC 20) sind nicht aufgeführt, aber dennoch voll im Basic 7.0 integriert**

## Musikalisches Basic

Ein ähnlicher Komfort ist auch bei der Programmierung des aus dem C 64 übernommenen Synthesizer-Bausteins, des SID, zu finden. Alle Musik-Parameter müssen nicht mehr aus DATA-Wüsten in den SID hineingePOKEt werden, sondern können elegant und leichtverständlich per Basic-Befehl gesetzt werden.

VOL regelt zum Beispiel die Lautstärke, mit dem SOUND-Kommando wird einer der Tongeneratoren gestartet. Dabei kann über entsprechende Parameter nicht nur die Frequenz, sondern auch die Dauer des Tones sowie das an- und abschwellen festgelegt werden.

Mit ENVELOPE wird jeweils eine von zehn möglichen Tonhüllkurven für Musikinstrumente definiert. Attack, Decay, Soustain, Release werden damit ebenso festgelegt wie Wellenform und Impulsbreite. Jede der zehn möglichen Hüllkurven bleibt gespeichert, bis sie durch einen weiteren ENVELOPE-Befehl zur gleichen Hüllkurvennummer überschrieben wird.

Nach dem Einschalten des PC 128 sind bereits alle zehn Hüllkurven mit der Klangstruktur verschiedener Musikinstrumente vordefiniert: Klavier, Akkordeon, Zirkusorgel, Trommel, Flöte, Gitarre, Cembalo, Orgel, Trompete und Xylophon. Damit steht auch dem musikalisch wenig bewandertem Einsteiger sofort eine Fülle einfach anwendbarer Klangeffekte zur Verfügung. Mit der FILTER-Anweisung können zudem alle Filtermöglichkeiten des SID zur

Klangverfremdung ausgeschöpft werden.

Der PLAY-Befehl ermöglicht das automatische Abspielen von in Strings gespeicherten Musiknoten. In dem als Parameter angegebenen String können Informationen über Hüllkurve, Oktave, Lautstärke, Tonkanal und Filter enthalten sein, in der Hauptsache aber natürlich die zu spielenden Noten. Die Noten werden einfach durch Angabe des Notennamens (A,B,C,D,E,F,G) ausgewählt, wobei das B der in Deutschland üblichen Notenbezeichnung H entspricht. Natürlich können die einzelnen Noten um Halbtöne erhöht oder erniedrigt werden, es sind ganze, halbe, viertel, achtel und sechzehntel Noten, jeweils auch punktiert, möglich.

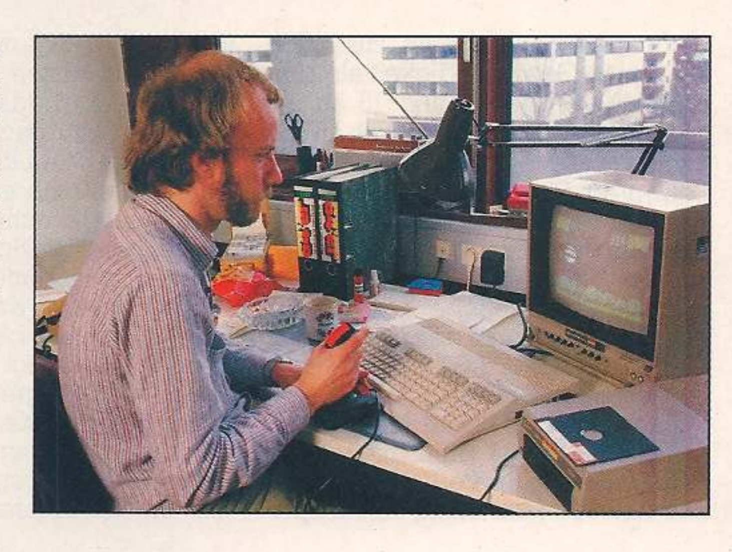


**Bild 13. Christian Rogge testete die Kompatibilität des PC 128 zum C 64. Selbst undefinierte Opcodes bereiteten dem PC 128 im 64er-Modus keinerlei Probleme.**

## Musik per Warteschlange

Auch der PLAY-Befehl wird interruptgesteuert ausgeführt, das heißt die zu spielenden Noten gelangen in eine Ton-Warteschlange, was nichts anderes bedeutet, daß sie in einem reservierten Speicherbereich abgelegt werden. Während des Interrupts stellt das Betriebssystem fest, ob Tondaten in der Warteschlange stehen. Wenn ja, wird der erste Wert aus der Schlange geholt und, vereinfacht gesprochen, an den Synthesizer-Chip (SID) zum Abspielen übergeben. Alle anderen in der Warteschlange stehenden Tondaten rücken jetzt einen Platz vor. Bei jedem weiteren Interrupt wird überprüft, ob die vorgesehene Tondauer bereits erreicht ist. Wenn dies schließlich der Fall ist, wird wieder nach wartenden Tondaten Ausschau gehalten, und das ganze Spiel setzt sich fort.

Wie gesagt läuft dies alles jeweils während des System-Interrupts ab. Das Basic selbst »merkt« davon nichts. Es stellt nur fest, ob noch Platz in der Tonwarteschlange ist oder nicht. Falls noch Plätze frei sind, können weitere Tondaten angefügt werden und das Programm fährt anschließend normal fort, während die Musik automatisch abgespielt wird. Nur dann, wenn zuviele Töne zum Abspielen anstehen, muß das Programm tatsächlich anhalten und warten, bis wieder Plätze in der Warteschlange freigeworden sind.

Obwohl Dank des leistungsstarken Basics nur selten nötig, gibt es natürlich auch beim PC 128 den Zugriff auf die Maschinenebene. Allerdings ist es hier nicht einfach mit POKE, PEEK und SYS getan. Vielmehr ergibt sich aus dem Konzept der verschiedenen Speicherbereiche, die mittels Bank-Switching umgeschaltet werden, das Problem, in welche Speicherbank der POKE-, PEEK- oder SYS-Befehl gehen soll. Das PC 128-Basic löst dieses Problem ebenso einfach wie elegant: Mit dem BANK-Befehl kann die gewünschte Speicherbank ausgewählt werden. Damit erfolgt natürlich nicht wirklich eine Bankumschaltung (während ein Basic-Programm läuft, muß natürlich immer der Basic-Interpreter eingeschaltet sein), sondern Basic merkt sich nur, in welcher Speicherbank beispielsweise ein POKE-Wert abgelegt werden muß, oder aus welcher Bank die Daten für PEEK stammen müssen.

## Die Verbindung zur Maschinensprache

So kann man nach Belieben entweder in den Programm- oder in den Variablenspeicher POKEn und PEEKen. Betriebssystem- und Basic-7.0-Routinen können nach »BANK 15« einfach mit SYS aufgerufen werden.

Daneben gibt es noch die Möglichkeit, von Basic aus ganze Speicherbereiche zwischen Bank 1 (Basic-Arbeitsspeicher) und anderen Speicherbänken hin- und herzuladen. Hierzu dienen die Befehle FETCH, STASH und SWAP. »FETCH 2000,50000,4,35000« holt beispielsweise 2000 Byte ab Adresse 35000 aus Speicherbank 4 und legt diese ab Adresse 50000 im Basic-Arbeitsspeicher (immer Bank 1) ab. STASH ist die Umkehrfunktion zu FETCH: Es wird eine Anzahl Bytes aus dem Arbeitsspeicher in eine andere Speicherbank gebracht. SWAP schließt tauscht die angegebenen Speicherbereiche in beiden Bänken gegeneinander aus.

Diese drei Befehle sind hauptsächlich für den Einsatz im Zusammenhang mit Speichererweiterungen (RAM-Floppy) gedacht.

Es können damit Datenmengen verwaltet werden, die ein mehrfaches von 64 KByte im Speicher belegen, und das mit Geschwindigkeiten, wie sie mit einer Floppy niemals zu realisieren sind.

## Maschinensprache-Monitor eingebaut

Wem trotz allem die Möglichkeiten des 7.0-Basic noch nicht reichen, der kann mit dem MONITOR-Kommando das Basic verlassen und landet im fest im ROM eingebauten Maschinensprachemonitor.

Dieser dem C 16-»Tedmon« nachempfundene Monitor enthält neben den üblichen Funktionen zum Listen und Beschreiben des Speichers und einem Disassembler auch einen kleinen Assembler, mit dem Maschinenprogramme sehr komfortabel eingegeben werden können.

Statt der sonst üblichen vierstelligen hexadezimalen Adresseneingabe verlangt dieser Monitor allerdings fünf Stellen:

Die erste Stelle gibt an, welche von 16 möglichen Speicherbänken ausgewählt werden soll. Allerdings sind in der Grundversion des PC 128 natürlich nicht alle 16 Bänke belegt, einige sind für ROM-Module, andere für die RAM-Floppy reserviert.

Zurück ins Basic gelangt man mit dem X-Kommando. Und wer schließlich genug hat von komfortabler Basic-Programmierung und lieber wieder mit POKEs und DATAs arbeiten will, dem steht schließlich für alle Fälle noch der C 64-Modus offen: GO 64.

(ev/hm)



# So lernt Ihr Drucker lesen

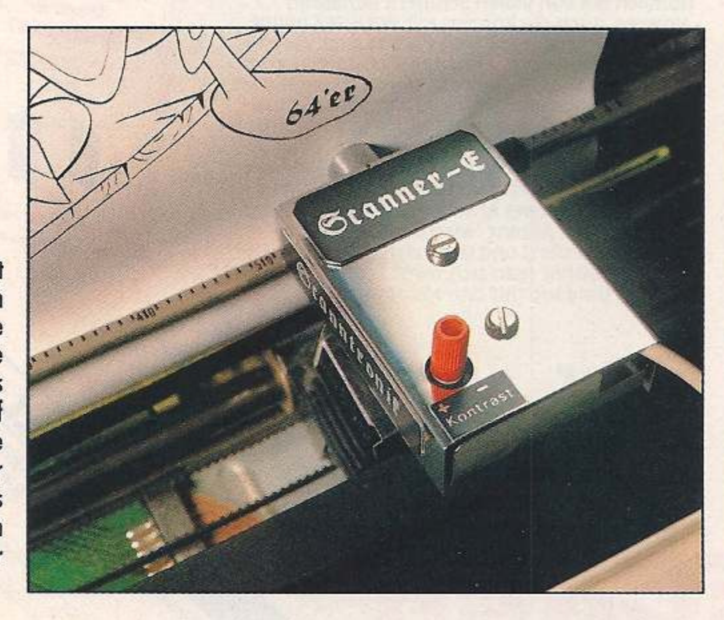
**Die Zeit mühsamer Grafikpinselei ist vorbei: Eine Hardware-Erweiterung für Epson-Drucker macht aus Ihrer Computer-Anlage den billigsten Scanner den es je gab. Beliebige Vorlagen werden damit in wenigen Minuten digitalisiert, und zur Weiterverarbeitung dem Anwender zur Verfügung gestellt.**

Der Druckerzusatz von Scanntronic besteht aus einer Fotozelle, die zusammen mit der benötigten Elektronik in einem kleinen Gehäuse untergebracht ist. Eine einfache Federklammer hält den »Abtaster« auf dem Druckkopf eines Epson RX-80F/T fest. Die Verbindung zum C 64 wird über den User-Port und das Druckerinterface hergestellt. Die mitgelieferte Software steuert die Bewegung des Druckkopfes und das Abtasten der Vorlage auf schwarz und weiß. Der Abtaster wird dazu mit dem Druckkopf horizontal über die Vorlage geführt und eine Punktreihe »gelesen«. Danach erfolgt ein punktweiser Zeilenvorschub — das Papier wird um 1/256 Zoll vertikal weiterbewegt — und die nächste Zeile wird gescannt. Eine präzise Druckermechanik ist also sehr wichtig.

**Mit wenigen Handgriffen kann ein Epson RX-80F/T in den billigsten Scanner den es je gab, umgerüstet werden. Versionen für andere Drucker sind in Vorbereitung und demnächst erhältlich.**

## Auflösung: 180000 Punkte

Ein Abtastpunkt hat einen Durchmesser von 0,25 mm. Diese Auflösung erlaubt sogar das »Lesen« von Schreibmaschinentext und normalgroßer Handschrift. Zum Abtasten können zwei Auflösungsstufen gewählt werden, die sich durch die Punktdichte und Bildgröße unterscheiden. In der Standardauflösung können Vorlagen der Größe DIN A5 quer und bei Hochauflösung Vorlagen der Größe DIN A6 hoch erfaßt werden. Pro Bild werden dabei etwa 180 000 Punkte abgetastet. Damit der Motivkontrast voll genutzt wird, kann der Scanner mit einem Regler an die Hell- und Dunkelwerte der Vorlage angepaßt werden.




**Die Optoelektronik ist in einem kleinen Metallgehäuse untergebracht. Eine Federklammer hält es auf dem Druckkopf fest. Links der rote Drehknopf zur Anpassung des Scanners an den Motivkontrast der Vorlage.**

Anstelle des Druckerpapiers spannen Sie beispielsweise eine Kopie von einem Malbuch in den Drucker ein und wählen den Programmpunkt »Bilderfassung«. Kaum 12 Minuten später ist das Bild in den Computer übertragen und kann beispielsweise mit HI-EDDI (64'er, Ausgabe 1/85) bearbeitet werden. Die Steuersoftware erlaubt auch das Speichern und Laden der Bilder auf Diskette. Ebenso können die Grafiken auf dem Bildschirm betrachtet werden. Da der Scanner mit einer höheren Auflösung arbeitet als der C 64, wird eine Grafik über vier Bildschirmseiten verteilt. Man sieht genau, welche Punkte bei einer späteren Hardcopy gesetzt sind. Das Einsatzspektrum des Scanners ist vielfältig; es reicht vom Fernkopierer bis zum Malprogramm-Ersatz.

## Einfache Handhabung

Zusammen mit einem Akustikkoppler können Sie Ihre Computeranlage als Fernkopierer einsetzen, um Skizzen, Schaltpläne oder handschriftliche Texte zu übertragen. Dazu spannen Sie die Vorlage in den Drucker ein und speichern die

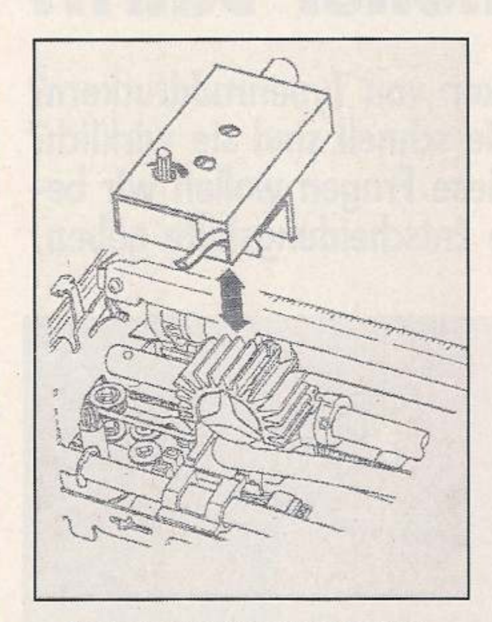
Hell/Dunkelwerte auf Diskette. Mit einem geeigneten Terminalprogramm können dann die Daten übers Telefonnetz gesendet werden. Der Empfänger kann sich dann die Grafik auf dem Bildschirm ansehen und ausdrucken lassen. Vorausgesetzt er hat die nötige Software.

Eine andere Anwendung ist das Anfertigen von Bildschirmgrafiken, das zu einem Kinderspiel wird. Das langwierige und zittrige Zeichnen mit dem Joystick oder Lichtgriffel gehört der Vergangenheit an.

## Scanner statt Joystick

Anstelle dessen greift man auf die bewährte Methode mit Bleistift und Papier zurück und digitalisiert einfach die Zeichnung. Wie bereits erwähnt ist eine Weiterbearbeitung der Grafiken mit HI-EDDI möglich. Gegenüber bekannten Malprogrammen mit Joystick-Steuerung reduziert sich dadurch die Zeit für das Erstellen von Grafiken um ein Vielfaches.

Mit einem kleinen Basic-Programm können Sie die gespeicherten Bilder laden und in eigenen Spielen verwenden. Im professionellen Bereich soll mit entsprechender Software sogar eine Auswertung von Fragebögen und Barcode-Symbolen möglich sein.



Hardcopy einer gescannten Vorlage. Eine beachtliche Qualität. Technische Zeichnungen können so einfach in Bedienungsanleitungen integriert werden.

Doch damit sind die Anwendungsmöglichkeiten noch lange nicht erschöpft: Ein denkbar fantastischer Einsatz wäre es, den Scanner zum Lesen von Listings herzunehmen.

Sie fotokopieren beispielsweise die Listings aus dem 64'er, spannen die Kopien in den Drucker ein und scannen sie. Ein Schrifterkennungs-Programm übernimmt dann die Auswertung der Grafik und erzeugt das Programm im Speicher des C 64 und auf Diskette oder Kassette.

Der Scanner arbeitet mit dem Epson Drucker RX-80F/T in Verbindung mit einem Wiesemann- oder Data-Becker Interface. Ausführungen für weitere Drucker- und Interfacetypen sind in Vorbereitung.

Zum Lieferumfang gehört die Optoelektronik, eine Programmdiskette sowie eine deutsche Bedienungsanleitung. Für den Preis von 348 Mark ist der Superscanner von Scanntronik ein sehr interessantes Gerät. Und das, wegen der vielseitigen Verwendbarkeit, nicht nur für den Computer-Grafiker. (hm)

Bezugsadresse: Scanntronik, Parkstr. 28, 8011 Zorneding, Tel. 08106/22570; Preis: 348 Mark



# Hammerwerke mit sehr schöner Schrift

**Wo liegen die Stärken von Typenraddruckern? Was können sie und wie schnell sind sie wirklich? Lohnt sich der Kauf? Diese Fragen wollen wir beantworten und Ihnen eine Entscheidungshilfe geben.**

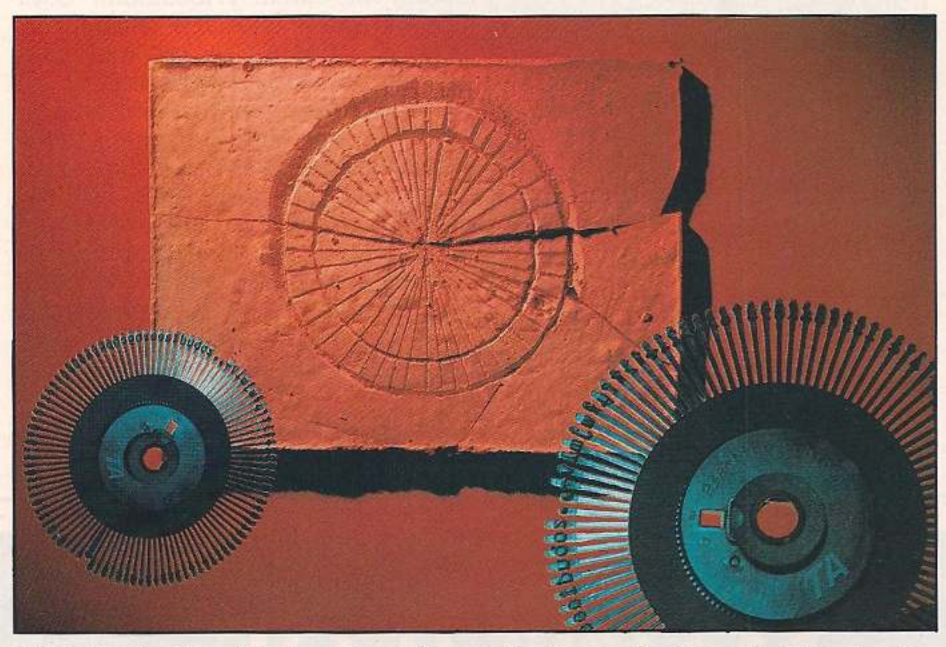


**Bild 1. Von der Lautstärke her erinnern Typenraddrucker an die altrömische Methode mit Meißel und Hammer.**

In der letzten Ausgabe haben wir Matrixdrucker unter die Lupe genommen. Diesmal sind die Schönschreiber, die Typenraddrucker, an der Reihe.

Um es gleich vorwegzunehmen: Typenraddrucker sind eigentlich nur zur Textverarbeitung gut. Für Listingsdruck sind sie zu teuer und zu langsam. Für Grafik sind sie nur sehr bedingt geeignet.

## Unschlagbares Schriftbild

Zu den Typenraddruckern sollen hier auch die Typenradschreibmaschinen gezählt werden. Gerade die Schreibmaschinen, mit entsprechendem Interface ausgestattet, sind als Typenraddrucker im Heimbereich verbreitet. Das ist einerseits mit dem Preis und andererseits mit der Vielseitigkeit dieser Maschinen zu begründen. »Reine« Typenraddrucker waren bis vor einiger Zeit für den Heimgebrauch unerschwinglich, im Gegensatz zu den elektrischen Schreibmaschinen. Wobei die Schreibmaschinen noch den Vorteil haben, auch als »Schreibmaschine« verwendbar zu sein. Soll nämlich nur mal schnell ein kurzer Brief geschrieben oder eine Postkarte ausgefüllt werden, ist es sehr zeitraubend, wenn erst der Computer eingeschaltet und das Textverarbeitungsprogramm geladen werden muß. Für jemanden, der die DIN-Tastenanordung einer Schreibmaschine kennt und nicht mit dem »Adler-Suchsystem« (erst kreisen und dann abstürzen) tippt, ergibt sich mit den Schreibmaschinen-Druckern noch ein weiterer Vorteil: Mit einem geeigneten, bidirektionalen Interface können die Texte auch mit der Schreibmaschine in den Computer eingegeben und dort bearbeitet werden.

Die bekanntesten drei Eigenschaften eines Typenraddruckers sind: schön, langsam und laut. Die Eigenschaften können mit der Funktionsart relativ einfach begründet werden.

## Wie funktionieren Typenraddrucker?

Typenraddrucker besitzen, wie der Name schon sagt, ein Typenrad. Das ist eine Plastikscheibe mit 96 oder mehr Segmenten, auf denen die einzelnen Zeichen (Typen) untergebracht sind (Bild 1 und 2). Die Scheibe hat einen Durchmesser von 8,5 cm. Wird nun das »A« gedrückt, dreht ein Schrittmotor das Typenrad so, daß die »A«-Type an der Schreibstelle zu liegen kommt. Ein Elektromagnet schlägt dann einen Bolzen von hinten auf die Type, die das Farbband auf das Papier drückt, und dort ein »A« hinterläßt (Bild 3). Ein Schrittmotor ist im Prinzip ein Elektromotor, dessen Achse mit einem Stromstoß um einen bestimmten Winkel gedreht werden kann.

## Langsame Mechanik

Nach dem Einschalten eines Typenraddrucker, dreht die Steuerelektronik für den Schrittmotor das Typenrad so lange, bis eine Lichtschranke die »Nullposition« feststellt. Der Lichtstrahl geht dann durch eine Bohrung in der Plastikscheibe hindurch und trifft auf einen Fototransistor. Wird nun auf einer Schreibmaschine eine Taste gedrückt, dreht die Steuerelektronik über den Schrittmotor das Rad so, daß die entsprechende Type sich in Anschlagposition befindet. Die heutige Elektronik ist dabei so »intelligent«, daß immer die günstigste

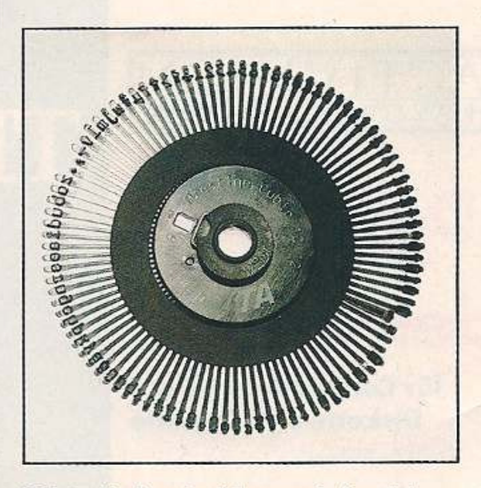


**Bild 2. Ein Standard-Typenrad. Deutlich sind die Typensegmente und die Bohrung für die Lichtschranke zu erkennen**

Drehrichtung (rechts oder links) gewählt wird.

Das Positionieren der Typen ist aber nicht nur ein elektronisches Problem, sondern erfordert auch mechanisch einen recht großen Aufwand. Das Typenrad muß nämlich sehr schnell gedreht und sehr exakt positioniert werden und der Bolzenhammer muß im richtigen Moment zuschlagen.

Um das nächste Zeichen zu drucken, muß die gesamte Mechanik einen Schritt weiter nach rechts transportiert werden. Die Mechanik ist dazu auf einem Wagen untergebracht, der auf einer Schiene nach links und rechts bewegt werden kann. Das Positionieren des Typenrades ist der geschwindigkeitsbestimmende Schritt. Die Kräfte, die beim Beschleunigen und Abbremsen auftreten, setzen die mechanischen Grenzen. Die Druckgeschwindigkeit liegt bei Typenraddruckern unter 2000 Mark in der Regel bei 10 bis 20 Zeichen pro Sekunde. Teure Exemplare schaffen mit dem entsprechenden mechanischen Aufwand auch 80 cps (character per second). Die Zeit für den Anschlag sollte aber auch nicht vernachlässigt werden. Der Bolzenhub, der nötig ist, um die Type auf das Papier zu schlagen, beträgt ein Vielfaches des Nadelhubs eines Matrixdruckers. Außerdem ist der Schlagbolzen noch wesentlich schwerer, womit der maschinengewehrartige Klang beim Druck erklärt werden kann.

| Anwendung | Matrixdrucker | Typenrad-drucker | Plotter |
|---|---|---|---|
| Text | gut | sehr gut | ungenügend |
| Listing | sehr gut | ausreichend | ungenügend |
| Grafik | befriedigend | ausreichend | sehr gut |
| Hardcopy | sehr gut | ungenügend | ungenügend |

**Tabelle 1. Übersicht der Einsatzgebiete der verschiedenen Druckertypen**

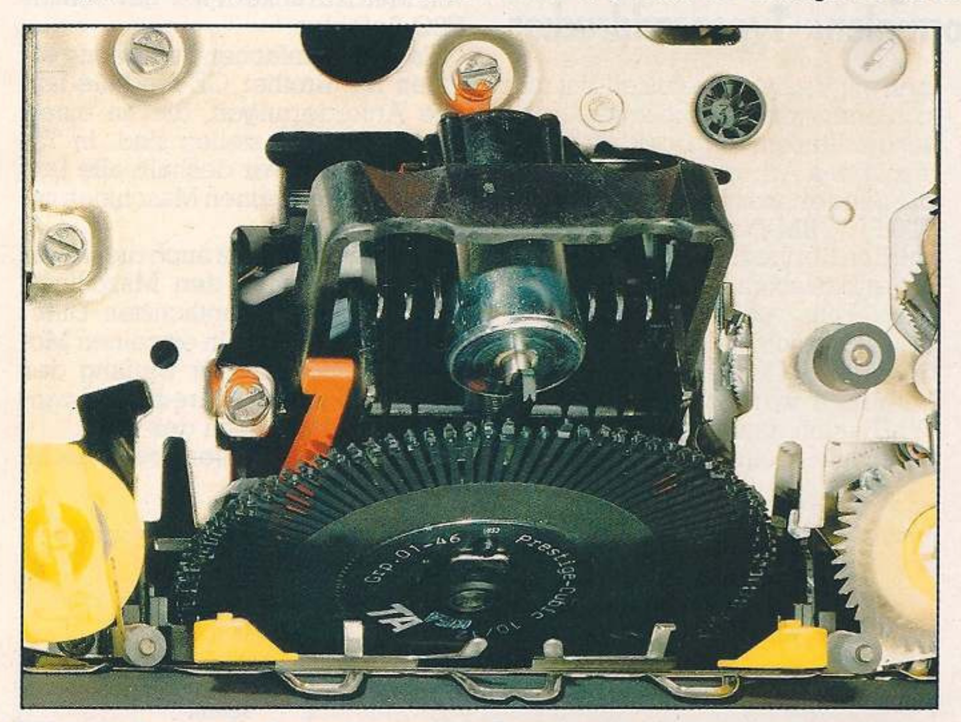


**Bild 3. Die Druckmechanik einer Typenradschreibmaschine. In der Mitte der Magnet zu sehen. Die ganze Einheit ist auf einem Transport-Schlitten untergebracht. Im unteren Teil des Bildes sehen Sie die Transport- und Anschlagwalze aus Gummi. Die gelben Rollen tragen das Korrekturband der Schreibmaschine.**

## Wozu ist ein Typenraddrucker geeignet?

Wegen der langsamen Druckgeschwindigkeit eignet sich ein Typenraddrucker nicht für Massen-Rundschreiben und Listingsdruck. Bei Listingsdruck können auch keine Steuer- oder Grafikzeichen des C 64 gedruckt werden. Teilweise werden solche Zeichen sogar ignoriert.

Bevor man also einen Drucker kauft, sollte man dessen Einsatzgebiet genau kennen. Tabelle 1 zeigt deshalb in knapper Form die Möglichkeiten der Druckertypen. Es wurde nicht nur das Ergebnis berücksichtigt, sondern auch die benötigte Zeit, um beispielsweise einen Grafikdruck zu erhalten.

Wie Sie aus der Tabelle ersehen können, ist der Matrixdrucker einer der Vielseitigsten. Er beherrscht fast alle anfallenden Druckprobleme mit recht guter Qualität. Berücksichtigen Sie aber, daß die Schrift billiger Matrixdrucker keine Briefqualität hat. Teure Stücke, etwa ab 1500 Mark, mit NLQ (Schönschreibmodus), kommen dem Schriftbild eines Typenraddruckers jedoch recht nahe.

Was Hardcopy-Fähigkeiten angeht, ist der Matrixdrucker mit Einzelnadelansteuerung unschlagbar. Einige Typenraddrucker sind zwar grafikfähig, benötigen aber für eine 320x200 Punkt Hardcopy zwischen einer halben und einer ganzen Stunde. Pro Grafikpunkt wird einmal das Punktzeichen angeschlagen. Also bis zu 64000mal pro Grafik — ein erheblicher Verschleiß für das Typenrad und die Anschlagmechanik. Außerdem ist dazu ein Programm erforderlich, das völlig anders als herkömmliche Hardcopy-Routinen aufgebaut sein muß. Zusätzlich muß ein Typenraddrucker für Hardcopy-Druck noch punktweisen Zeilen- und Spaltenvorschub ermöglichen, was nur selten der Fall ist.

## Eingeengtes Anwendungsspektrum

Sie sehen also, ein Typenraddrucker hat ein sehr eingeengtes Anwendungsspektrum. Unserer Meinung nach lohnt er sich nur, wenn es auf sehr schöne Schrift oder mehr als zwei Durchschläge ankommt. Etwas anders verhält es sich mit Schreibmaschinen, die als Drucker verwendet werden. Hat man doch neben einem Drucker noch eine komplette Schreibmaschine für kurze Texte zur Verfügung. Außerdem kann die Schreibmaschine auch als alternative Eingabetastatur verwendet werden.

(hm)

# Big Brother druckt für Sie

**Sehr gute Leistungen versprechen die Brother Schreibmaschinen CE 51 bis 61. Über ein Interface können sie an den C 64 angeschlossen werden und ersetzen dann einen »normalen« Typenraddrucker.**

Versuchen Sie einmal, ein Behördenformular mit einem Drucker auszufüllen. Sie werden sehen, es ist beinahe unmöglich genau die richtige Spalte und Zeile zu treffen. Für diesen Anwendungszweck sind Schreibmaschinen noch am besten geeignet. Gleichzeitig hat eine Typenradschreibmaschine meistens ein wesentlich besseres Schriftbild als ein Matrixdrucker. Aber warum sollte man nicht beide Vorteile miteinander verbinden können? Eine herkömmlich zu bedienende Schreibmaschine, die bei Bedarf von einem Computer gesteuert werden kann, bietet somit doppelten Nutzen. Natürlich hat diese universelle Einsetzbarkeit ihren Preis. Die drei Brother-Schreibmaschinen, die wir Ihnen vorstellen möchten, kosten 1 098 Mark (CE 51, Bild 1) 1 298 Mark (CE 60) und 1 398 Mark (CE 61, Bild 2). Dabei wurde ein in jedem Fall notwendiges Interface zum Anschluß an den Computer noch nicht mitgerechnet. Hierfür fallen nochmals zwischen 400 und 500 Mark, je nach verwendetem Modell an. Zum Test verwendeten wir das bidirektionale Witte WI-60 (549 Mark) und das unidirektionale IF-CBM-Interface von Brother (399 Mark).

#### C 64 mit neuer Tastatur

Gut ausgewogen ist der Tastenwiderstand, der ein Auflegen der Finger ermöglicht. Nur der Druckpunkt beim Betätigen einer Taste könnte etwas deutlicher sein.

Da mit dem WI-60 ein bidirektionaler Datenaustausch möglich ist, kann die Schreibmaschine auch als Eingabegerät genommen werden. Das funktioniert nicht nur mit Basic-Programmen, sondern sogar mit Vizawrite, Textomat und SM-Text. Für alle anderen Programme ist es möglich, sich eine eigene Tastaturtabelle zu erstellen, um die Tastenbelegung dem Programm anzupassen. Wir haben das WI-60 zusammen mit Vizawrite getestet. Hat man sich erst einmal an die unterschiedlichen Tastenfunktionen gewöhnt, macht das Arbeiten richtig Freude. Die Sondertasten des C 64 (F1 — F8, STOP, CBM-Taste, Cursor-Tasten) wurden übrigens sinngemäß auf die Schreibmaschinentastatur übertragen. Falls neben der Brother-Schreibmaschine noch ein Matrixdrucker mit Centronics-Schnittstelle betrieben werden soll, bietet das WI-60 auch dafür einen passenden Anschluß. In einigen Fällen kann es allerdings passieren, daß das eine oder andere Spiel nicht mehr funktioniert. In diesen Fällen muß das Interface aus dem Computer genommen werden.

#### Langsam aber bedienungsfreundlich

Die umfangreichen Funktionen aller drei Brother-Schreibmaschinen können auf zwei verschiedene Arten eingestellt werden. Entweder über die Schreibmaschine oder durch den Computer. Die Syntax dieser Befehle entspricht dabei den von Matrixdruckern her gewohnten ESC-Befehlen.

Trotz der einfachen Bedienung erfüllen die Brother CE-Modelle fast alle Anforderungen, die an einen Textdrucker zu stellen sind. In Tabelle 1 haben wir deshalb alle Leitungen der einzelnen Maschinen zusammengestellt.

Die Tabelle macht auch die Unterschiede zwischen den Maschinen deutlich. Die wesentlichsten Differenzen zwischen den einzelnen Modellen sind aber der Umfang der Ausstattung, die Schreibgeschwindigkeit und natürlich der Preis.

Das Schriftbild aller drei Maschi-

| Funktion | Brother CE-51 | Brother CE-60 | Brother CE-61 |
|---|---|---|---|
| Zeichenabstand | 10,12,15 Zeichen/Zoll | 10,12,15 Zeichen/Zoll | 10,12,15 Zeichen/Zoll Proportionalschrift |
| Zeilenabstand | 1/1,5/2 | 1/1,5/2 | 1/1,5/2 |
| Zeileneinrückung | Ja | Ja | Ja |
| Linken und rechten Rand setzen | Ja | Ja | Ja |
| Tabulatoren setzen | Ja | Ja | Ja |
| Dezimal-Tabulator | Ja | Ja | Ja |
| Halbzeilenschaltung | Ja | Ja | Ja |
| Manelle Korrektur | Ja | Nein | Nein |
| Halbzeilenschaltung Vor/Rück | Ja | Ja | Ja |
| Unterstreichautomatik | Nein | Ja | Ja |
| Rückwärtstabulator | Nein | Ja | Ja |
| Feinschrittaste | Nein | Nein | Ja |
| Taste für Fettschrift | Nein | Nein | Ja |
| Druckgeschw. Zeichen/sek. | 13 | 13 | 15 |
| Eingabepuffer | 48 Zeichen | 48 Zeichen | 48 Zeichen |
| Korrekturpuffer | 165 Zeichen | 165 Zeichen | 165 Zeichen |
| Zeit für Probetext (zum Vergleich FX 80 : 2:15) | 16:42 | 16:40 | 16:36 Min. |
| Preis (DM): | 1 098 | 1 298 | 1 398 |

**Tabelle 1. Die Funktionen der Brother-Schreibmaschinen CE-51, CE-60, CE-61**


Bild 1. Die einfachste, die CE-51 von Brother


Bild 2. Identisch vom Aussehen: CE-60 und CE-61

Die Brother Schreibmaschinen CE-51, CE-60, CE-61
Hervorragende Leistungsfähigkeit und gutes Design sind die Merkmale der Brother CE-51, CE-60, CE-61. Sie verfügen über ein exzellentes Schriftbild, das sich bestens zur Textverarbeitung eignet.

**Das Schriftbild der Fettschrift ist überzeugend**

Bild 3. Schriftprobe der Brother-Schreibmaschinen (Originalgröße)

nen kann eigentlich nur als exzellent bezeichnet werden (Bild 3). Verwendet man ein Carbonband, sind die Buchstaben wahrlich schöner als gedruckt. Besonders gut gefallen hat uns der Doppeldruck, der den Buchstaben ein räumliches Aussehen verleiht.

## Schrift ohne Makel

Je nach dem verwendeten Typenrad sind so die verschiedensten Schriftvariationen bis hin zur Proportionalschrift zu erreichen. Eines sollte der Besitzer einer Typenradschreibmaschine allerdings haben — nämlich Zeit. Mit 13 oder 15 Zeichen/Sekunde (CE-51, CE-60) dauert es schon eine ganze Weile, bis ein Schriftstück fertig ist. Einzige Abhilfe ist hier ein Druckpuffer, der möglichst groß sein sollte, damit der Computer nicht zu lange an den Drucker gebunden ist. Das WI-60-Interface hat beispielsweise einen 8 KByte, das IF-CBM-Interface einen 2 KByte (optional 6 oder 8 KByte) großen Pufferspeicher.

Da die Schreibgeschwindigkeit eines Druckers nicht nur von der Anzahl der geschriebenen Zeichen pro Sekunde abhängig ist, sondern auch von anderen Faktoren, haben wir einen über drei Seiten langen Probetext entwickelt, der in Zukunft bei allen Druckertests verwendet wird. Er ist so konzipiert, daß auch Druckwegoptimierung, Papiertransport, Positionierungsgeschwindigkeit des Typenrades und Druckkopfgeschwindigkeit mit berücksichtigt werden. Als Textverarbeitungsprogramm für diesen Geschwindigkeitstest dient Vizawrite.

Unsere drei Testkandidaten waren die ersten, die diese Teststrecke absolvieren mußten. Sie schafften es mit relativ guten Ergebnissen. Die gemessenen Werte entnehmen Sie bitte der Tabelle 1. Erstaunlich ist aber, daß alle drei Maschinen, trotz unterschiedlicher Schreibgeschwindigkeit fast genau gleich schnell waren. Der Unterschied zwischen der langsamsten und schnellsten Maschine betrug nur sechs Sekunden.

## Einfache Handhabung

Das Typenrad wird beim Einsetzen automatisch justiert. Der Farbbandwechsel (Carbon- oder Nylonband) ist so einfach, daß er nicht einmal 10 Sekunden dauert. Nur das Einsetzen des Korrekturbandes dauert etwas länger. Wer aber mit dem Computer schreibt, kann sich das Korrekturband ohnehin sparen, denn Fehler werden einfach am Bildschirm berichtigt.

**Problemloser Anschluß**

Das Brother IF-CBM-Interface ist mit dem ganz normalen C 64-IEC-Bus ausgestattet. Es wird einfach zwischen Computer (beziehungsweise Floppy-Laufwerk) und Schreibmaschine geschaltet. Damit fehlt dem Interface zwar die Fähigkeit bidirektional arbeiten zu können, dafür treten aber mit keinen käuflichen Softwareprodukten Probleme auf. Außerdem paßt das Interface auch zum VC 20 beziehungsweise dem PC 128. Die Schreibmaschinen werden einfach wie bei einem Commodore-Drucker über die Geräteadresse 4 angesprochen. Ein Funktionieren mit Vizawrite 64, SM-Text und Textomat ist sichergestellt. Sollten dennoch einmal Schwierigkeiten auftreten, bietet der Hersteller bereitwillig seine Hilfe an.

Die drei getesteten Geräte, aber auch die Interfaces von Witte und Brother, haben sich im Test durchweg gute Noten verdient. Je nach Anwendung (und Geldbeutel) kann sich der Interessierte mit ihnen ein leistungsfähiges und flexibles Textsystem zusammenstellen. Der Vorteil liegt vor allem darin, daß die bekanntesten Textverarbeitungsprogramme problemlos angewendet werden können.

(Arnd Wängler/hm)

Info: Brother, Rosengarten 14, 6368 Bad Vilbel, Tel. 06101/8050

# Rendezvous mit der Schreibmaschine

## Für knappe 1000 Mark ist die Triumph-Adler Gabriele 8008 L eine der billigsten Typenradschreibmaschinen mit eingebautem Commodore-Interface.




Bild 1. Eine solide Schreibmaschine

```
Schriftproben Gabriele 8008 1

Sehr gutes Schriftbild mit Prestige-Cubic

Sonderzeichen: µ, ², ³, £, °, /, <, >
```

Bild 2. Serienmäßig gute Schrift

```
100 REM STEUERZEICHEN MIT GABRIELE
110 :
120 PRINT"ª"        :REM CLR HOME
130 PRINT"î"        :REM RVS ON
140 PRINT"ĵ"        :REM RVS OFF
150 PRINT"â"        :REM CTRL+1
160 PRINT"ô"        :REM CTRL+2
170 PRINT"ê"        :REM CTRL+3
180 :
190 :
```

Bild 3. Die Steuerzeichen

Wer für 898 Mark die Triumph-Adler Gabriele 8008 L als Typenradschreibmaschine kauft, weil er eine Schreibmaschine braucht, kann tatsächlich für nur 100 Mark Zuzahlung die Maschine mit Interface zum Anschluß an die Commodore-Computer C 16, VC 20, C 64 und Plus/4 bekommen! Wenn man die Notwendigkeit der Anschaffung einer Schreibmaschine voraussetzt, ist das eine der billigsten Methoden, einen soliden Schönschreibdrucker für den Computer zu bekommen.

Die Typenradschreibmaschine (Bild 1) macht äußerlich einen robusten Eindruck. Geliefert wird sie in einem sehr stabilen Koffer, aber ohne Abdeckung. Auch sonst ist bei vielen nicht unbedingt nötigen Ausstattungsmerkmalen gespart worden. Der Papiereinzug erfolgt nur mechanisch und erfordert zunächst einige Übung, funktioniert dann aber problemlos. Es fehlen eine Halbschrittaste und ein Andruckregler.

Nach dem Einschalten herrscht wohltuende Ruhe, das bei manchen Konkurrenzmodellen zu hörende leichte Summen des Elektromotors ist nicht da. Leider vergibt die Maschine diesen Vorteil durch ein recht lautes Geräusch beim Schreiben. Ursache ist sicher die einfache Walze, auf die das Typenrad unbarmherzig schlägt. Unangenehm fällt zusätzlich ein Klacken vor dem Zurückkehren des Typenträgerwagens auf.

Das DIN-Tastenfeld enthält 46 Schreibtasten für 92 Zeichen. Weitere 8 Zeichen sind mit der Mehrfachbelegungstaste erreichbar. In Abhängigkeit vom verwendeten Typenrad kann in Picaschrift oder Eliteschrift (serienmäßig ist Prestige-Cubic, Bild 2) geschrieben werden. Fünf programmierbare Tabulatorstopps und ein Dezimaltabulator erleichtern die Arbeit. Beim Ausschalten werden die neuen Randpositionen und Tabulatoreinstellungen gelöscht. Auch große Umschläge und querliegende DIN-A4-Bogen lassen sich einspannen. Automatisches Unterstreichen und rechtsbündiges Schreiben sind leider nicht möglich. Typenrad und Farbbandkassette lassen sich leicht wechseln.

Die Gabriele 8008 L wird mit V.24-Schnittstelle und mit VC-Interface zum Anschluß an Commodore-Computer angeboten. In der getesteten Ausführung war das VC-Interface eingebaut. Verbunden werden Computer und Gabriele 8008 L mit einem 120 cm langen Kabel, das im Lieferumfang enthalten ist. Der Drucker wird über die Geräteadresse 4 angesprochen. Aber auch eine Änderung auf Adresse 5 ist möglich. Durch die Sekundäradressen können Cursor-up-, Cursor-down- und Original-Modus eingestellt werden. Die 8 zusätzlichen Zeichen der Schreibmaschine lassen sich leider nicht ansteuern. Selbstverständlich können auch die Grafikzeichen und die Steuerzeichen nicht ausgedruckt werden. Für den persönlichen Gebrauch haben die Konstrukteure aber für die Kontrollcodes im Cursor-up-Modus eine recht brauchbare Ersatzlösung gefunden (Bild 3). Warum fehlt eine entsprechende Lösung für die Cursorsteuerzeichen? Es wird nicht einmal ein Zwischenraum für nachträgliche Einfügungen ausgedruckt.

Zu beanstanden ist auch, daß ein Ändern der Zeilenlänge, genauer gesagt des rechten Randes, nicht vom Computer aus möglich ist. Das hat schlimme Folgen, wenn man versucht, über diesen Rand hinweg drucken zu wollen: Diese Zeichen werden einfach verschluckt. Das gilt auch für Listings, deren Basic-Zeilen länger als 65 Zeichen sind. Abhilfe schafft man nur, indem der rechte Rand manuell über die Schreibmaschinentastatur geändert wird. Die Gabriele 8008 L zeigt hier also das typische Verhalten einer Schreibmaschine.

Ansonsten arbeiten Schreibmaschine und Interface mit den gängigen Textverarbeitungsprogrammen problemlos zusammen. Natürlich kann das Interface nicht mehr als das Textverarbeitungsprogramm. Wenn die Umlaute im Programm fehlen, muß man eben darauf verzichten!

Zu den schnellen Druckern gehört die Gabriele mit 10 Zeichen pro Sekunde ganz sicher nicht. Entsprechend seiner Herkunft von einer Schreibmaschine ist dem Drucker der bidirektionale Druck verwehrt. Eine engbeschriebene Seite im DIN-A4-Format braucht so schon ihre zehn Minuten, genug Zeit für eine Kaffeepause im Nebenraum, um so dem lästigen Lärm zu entgehen. Von Nachteil ist auch, daß das Interface nicht über einen Puffer verfügt. Der Computer ist also für die Zeit des Druckens blockiert!

Zusammengefaßt läßt sich sagen, daß die Gabriele 8008 L mit Interface wegen der doppelten Nutzbarkeit als Schreibmaschine und als Schönschreibdrucker ein sehr günstiges Preis/Leistungsverhältnis bietet. Die aufgeführten Komfortmängel kann man unter Berücksichtigung des relativ niedrigen Preises und des hervorragenden Schriftbildes akzeptieren.

(Dieter Hein/gk)

Info: Triumph-Adler AG, Fürther Straße 212, 8500 Nürnberg, Tel. 0911/333-0

# Uchida DWX 305 — Klappern gehört zum Handwerk

**Der Typenraddrucker, der nicht mehr zu hören ist, muß erst noch erfunden werden. Dennoch überrascht der DWX 305 mit relativ niedrigem Geräuschpegel und einer Schriftqualität, die jenseits von Gut und Böse liegt.**

Den Eindruck, den der DWX 305 (Bild 1), nicht nur auf dem Papier, hinterläßt, ist bestechend. Kein Wunder, denn als Typenraddrucker braucht er sich im Gegensatz zum Matrixdrucker keine Gedanken darüber zu machen, wie ein Buchstabe aussieht. Die ihm zur Verfügung stehenden 96 Zeichen sind alle auf einem runden Typenrad angeordnet. Zusammen mit dem nur einmal verwendbaren Carbonband aus der Farbbandkassette (zirka 25 Mark) verleihen sie dem DWX 305 eine gestochen scharfe Schrift (Bild 2).

## Der Spezialist

Als typischer Spezialist hat der DWX 305 seine Stärken und Schwächen. Während Schönschrift seine Domäne ist, sind ihm Begriffe wie Grafik oder Zeichnen fast vollkommen unbekannt. Er wurde für die Textverarbeitung konstruiert. Dafür ist er allerdings bestens ausgestattet. Er verarbeitet sowohl Papier im DIN-A4-Längsformat, als auch solches mit einer Breite bis zu 33 Zentimetern (DIN A4 quer). Auf dieser Papierbreite lassen sich, je nach Zeichendichte, 120, 144 oder sogar 180 Zeichen unterbringen. Leider bleibt es dem Anwender überlassen, dem Drucker mitzuteilen, wann er in eine neue Zeile springen soll. Verwendet man nämlich Papier, das kleiner als die Druckwalze ist, kann es leicht passieren, daß der Druckkopf plötzlich ins Leere hämmert. Es gibt beim DWX 305 keine Möglichkeit, die Papierbreite einzustellen. Manche andere Drucker verwenden hierzu einen speziellen Schalter oder besitzen zumindest einen Steuercode zum Einstellen des rechten Randes. Die praktikabelste Lösung dieses Problems ist es, schon beim Anfertigen des Textes darauf zu achten, nicht mehr Zeichen zu schreiben, als auf dem Papier unterzubringen sind. Wird der DWX 305 zusammen mit einem Textverarbeitungsprogramm eingesetzt (was der Regelfall ist), braucht man sich darüber glücklicherweise keine Gedanken machen, denn die Kontrolle der Textbreite wird vom Programm übernommen. Die Druckgeschwindigkeit beträgt dabei höchstens 18 Zeichen pro Sekunde, wenn keine Sonderfunktionen wie Fettdruck, Schattenschrift oder das Unterstreichen gewählt werden.

## Drucker für Geduldige

Mischt man allerdings mehrere dieser Druckfunktionen, beispielsweise, um wichtige Textpassagen hervorzuheben, reduziert sich die Druckgeschwindigkeit erheblich. Da bei einem Zeichen dann bis zu viermal angeschlagen wird, braucht man sehr viel Geduld. Zum Vergleich: Ein guter Matrixdrucker schafft in der Regel mehr als hundert Zeichen pro Sekunde, allerdings mit meist wesentlich schlechterem Schriftbild. Als Papierantrieb dient dem DWX 305 in der Standardausführung ein Friktionsantrieb (Papiertransport durch Gummiwalze). Zusätzlich erhältlich sind ein Einzelblatteinzug (799 Mark) und ein aufsetzbarer Traktor für Papier mit Lochrand (299 Mark). Der Einbau beider Zusatzgeräte ist, ebenso wie die gesamte Handhabung des Druckers, einfach und kann auch vom technisch Ungeübten vorgenommen werden. Damit eine Fehlbedienung ausgeschlossen ist, be-



**Bild 1. Uchida DWX 305: Ein preiswerter Typenraddrucker**

```
Der UCHIDA DWX-305 am Commodore 64

Der DWX-305 ist ein komfortabler, reichhaltig
ausgestatteter Typenraddrucker mit:

-Deutschen Umlauten: Ö Ä Ü ö ä ü ß, und
+verschiedenen Schriften:
-Der Fettdruck dient zum Hervorheben.
-Mit dem Schattendruck sind reizvolle Effekte möglich.
-Ein Mischen verschiedener Modi ist sehr einfach

+Eine Besonderheit sind die vielen Formatbefehle der Textausgabe.
+Der DWX-305 verwendet Typenräder und Farbbänder der Qume-Drucker.
```

**Bild 2. Der DWX 305 beherrscht die wichtigsten Schriftarten**

finden sich am DWX 305 zwei Schalter und eine Lichtschranke auf dem Druckkopf. Es ist somit schwer möglich, mit falsch eingelegtem Papier, geöffnetem Gehäusedeckel oder verbrauchtem Farbband zu arbeiten. Der Drucker zeigt alle Fehlfunktionen durch die ERROR-Taste auf der Gehäusevorderseite an. Mit ihr macht man den Drucker nach einer Störung auch wieder betriebsbereit. Neben der ERROR-Taste und den drei üblichen Tasten für ON/OFF-Line, Zeilen- und Seitenvorschub findet man auch eine Taste mit dem Namen SET PAGE. Mit ihr kann der Beginn einer Seite neu eingestellt werden (das interne Zeilenzählregister wird gelöscht). Der Selbsttest des DWX 305 wird durch eine separate Taste im OFF-Line-Modus ausgelöst.

## Anschluß am C 64

Die Hersteller haben den DWX 305 ab Werk mit einer Centronics-, oder wahlweise mit einer RS232-Schnittstelle ausgestattet. Für den Anschluß an den C 64 empfiehlt sich die Centronics-Version. Zum einen ist sie mit 1349 Mark etwas billiger als die serielle Version (1499 Mark). Zum anderen genügt ein einfaches Userport-Kabel und eine Software-Schnittstelle, wie sie beispielsweise in Vizawrite 64 einprogrammiert ist. Wer den DWX 305 gerne am seriellen Port betreiben möchte, kann sich mit dem einfachsten Interface begnügen, denn Sonderfunktionen wie Grafik oder eine Umsetzung der Commodore-Steuerzeichen sind bei einem Typenraddrucker ohnehin sinnlos. Trotzdem stehen dem Besitzer eines DWX 305 eine Reihe verschiedener Schriften zur Verfügung — durch Austausch des Typenrades. Zur Auswahl steht ein weites Spektrum verschiedenster Schriftarten, denn die weit verbreiteten Qume-Typenräder passen ausgezeichnet.

Obwohl der Befehlssatz des DWX 305 nicht gerade üppig ist, besitzt er, mit Ausnahme der rechten Randeinstellung, alle wichtigen Steuercodes. In Verbindung mit manchen Textverarbeitungsprogrammen ist sogar ein Ausdruck in proportionalem Zeichenabstand möglich, denn die vertikale Druckposition eines Zeichens läßt sich vom Computer aus genau festlegen. Eine bisher bei anderen Druckern wenig verbreitete Besonderheit des DWX 305 ist seine variable Anschlagstärke. Durch zwei DIL-Schalter auf der Gehäuserückseite kann zwischen weichem, mittleren und hartem Anschlag selektiert werden. Damit wird nicht nur die Geräuschentwicklung des Druckers maßgeblich beeinflußt, sondern auch die Qualität der drei möglichen Durchschläge.

## Profi für zu Hause

Obwohl der DWX 305 sicherlich auch manchem kleineren Unternehmen bei seinen täglichen Schreibarbeiten dienlich sein könnte, macht ihn sein relativ günstiger Preis (1349 Mark inklusive MwSt) auch für die private Nutzung interessant. Dabei sollte natürlich beachtet werden, daß auch der DWX 305, wie alle Typenraddrucker, zur ausschließlichen Textwiedergabe geeignet ist. Wer nicht all zu hohe Ansprüche an die Schreibgeschwindigkeit stellt, findet im DWX 305 einen zuverlässigen Drucker zu günstigem Preis.

(Arnd Wängler/gk)

Info: Synelec Datensysteme GmbH, Postfach 151727, 8000 München 15, Tel. 089/7253081

# Marktübersicht: Typenraddrucker und Schreibmaschinen

**Die »Schönschreibdrucker« werden ständig billiger. Grund für uns, Ihnen mit einer Marktübersicht eine Einkaufshilfe zu geben.**

Typenraddrucker funktionieren ähnlich einer elektrischen Schreibmaschine — einige sind sogar komplette Schreibmaschinen. Über eine Magnetspule mit einem Bolzen werden die einzelnen Typen aufs Papier geschlagen. Eine laute und langsame Angelegenheit also. Die Schriftqualität ist aber dafür hervorragend. Auch sind mehr Durchschläge als bei Matrixdruckern möglich.

**Worauf sollte man achten?**

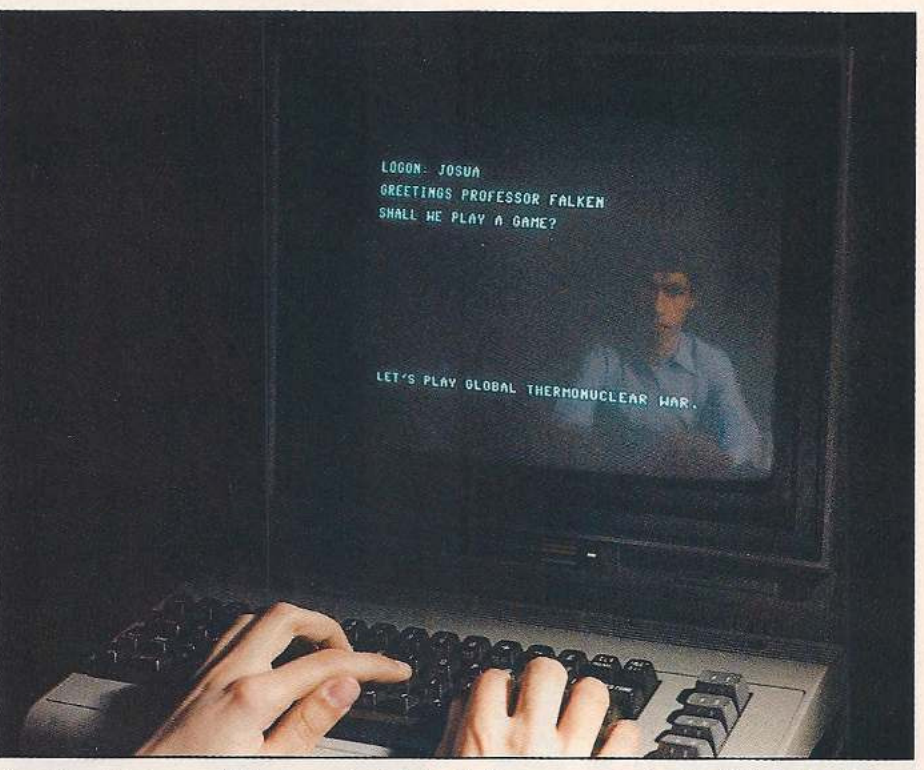
Einmal auf das Einsatzgebiet des Druckers. Ein Typenraddrucker lohnt sich erst dann, wenn Sie ständig Briefe mit Ihrem Computer schreiben wollen und die Korrespondenz über ein ausgezeichnetes Schriftbild verfügen soll. Denn für den Ausdruck nur von Listings sind diese Drucker zu teuer und für HiRes-Bilder nur bedingt geeignet. Meist verwenden sie sogar ein Plastik-Carbonband, das nach einem Durchlauf gegen ein neues ersetzt werden muß. Der Druck einer Punktgrafik des C 64 dauert mit einem Typenraddrucker etwa eine halbe bis eine ganze Stunde (320x200 Punkte). Vorausgesetzt, punktweiser Zeilen- und Spaltenvorschub ist möglich und man hat ein entsprechendes Programm. Die Abnutzung der Punkt-Type dürfte dabei allerdings sehr groß werden. Zusammenfassend kann man sagen, daß Sie mit einem Typenraddrucker richtig beraten sind, wenn Sie fast ausschließlich Texte mit ausgezeichneter Qualität drucken wollen.

In der folgenden Marktübersicht stellen wir Typenraddrucker und Schreibmaschinen vor, die sich auch als Drucker eignen. Die Preisgrenze haben wir bei 2000 Mark inklusive Interface gezogen. (hm)

| a) Anbieter<br>b) Hersteller | Modell | Grafikfähig | Druckgeschwindigkeit Zeichen pro Sekunde | Antrieb: Walze (1), Traktor (2) | Papierbreite (cm) | Durchschläge inkl. Original | Lautstärke in dB (A) | Druckpuffer (in KByte) | a) Abmessungen (HxBxT) in cm<br>b) eigene Tastatur | Schnittstellen: RS232 = 1, Centronics = 2, C 64 = 3 | a) Druckerkabel im Lieferumfang enthalten<br>b) anschlußfertig an C 64<br>c) inkl. Interface | Empfohlenes Interface | Preis inkl. MwSt. (DM) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a) **Brother International**<br>b) Brother International | CE-51 | nein | 13 | 1 | 34,3 | 5 | <65 | 4 | a) 13,8x 43,4x39,3<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | IF-CBM 399,– | 1098,– |
| | CE-60 | nein | 13 | 1 | 34,3 | 5 | <65 | 4 | a) 13,8x 43,4x39,3<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | w. o. | 1298,– |
| | CE-61 | nein | 15 | 1 | 34,3 | 5 | <65 | 4 | a) 13,8x 43,4x39,3<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | w. o. | 1398,– |
| | HR-15XL | ja | 18 | 1, 2 | 34,3 | 4 | <65 | 5 | a) 16,6x 46,4x33,0<br>b) ja | 1, 2 | a) nein<br>b) nein<br>c) ja | k. A. | 1824,– (Centr.) |
| a) **Microscan**<br>b) Microscan | MS-15VC | nein | 15 | 1, 2 | 34 | 5 | <65 | 1 Zeile | a) 16x 49,5x32,2<br>b) nein | 1, 2, 3 | a) ja<br>b) ja<br>c) ja | k. A. | 1698,– |
| a) **Mirwald Electronic**<br>b) Mirwald-Petal | PETAL-MA20 | ja | 18 | 1 | 33 | 4 | <63 | 2 | a) 15,1x52 x36<br>b) nein | 1, 2, 3, IEEE-488 | a) nein<br>b) nein<br>c) Aufpreis | C 64-Interf. 249,– | 1498,– |
| a) **NEC Business Systems**<br>b) NEC Corporation Tokio | Spin-writer Elf - 360 | ja; 48 x120 | 19 | 1, 2 | 30 | 4 | 55 | 3,5 | a) k. A.<br>b) nein | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | Wiese-mann | 1664,40 |
| a) **Olympia**<br>b) Olympia | Com-pact2 | nein | 14 | 1 | 36,4 | 6 | k. A. | 172 Byte | a) 13,7x 47,6x39,8<br>b) ja | 1, 2 | a) nein<br>b) nein<br>c) nein | k. A. | 1480,– |
| a) **Quen Data**<br>b) Quen Data | Excellen-ce 70 VC | nein | 12 | 1 | 34 | 5 | k. A. | k. A. | a) 14x 45x47<br>b) ja | 3 | a) nein<br>b) ja<br>c) ja | k. A. | 998,– |
| | Executi-ve 80 | nein | 15 | 1 | 30,5 | 5 | k. A. | k. A. | a) 17x50 x50<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | IFU 2 A | ab 1298,– |
| | DWP 1120 | nein | 18 | 1 | 33 | 6 | 58 | k. A. | a) 17,1x 53,2x34,2<br>b) nein | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | IFU 2 A | 1198,– |
| a) **Silver Reed Int.**<br>b) Silver Seiko Ltd. | EX 42 Commo-dore | nein | 12 | 1 | 30,5 | 4 | 65 | k. A. | a) 14x43 x36<br>b) ja | 3 | a) ja<br>b) ja<br>c) ja | k. A. | 1251,72 |
| | EX 43 | nein | 12 | 1 | 30,5 | 4 | 65 | k. A. | a) 14x43 x36<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | k. A. | 963,30 |
| | EX 43N | nein | 12 | 1 | 30,5 | 4 | 65 | k. A. | a) 14x43 x36<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | k. A. | 1023,72 |
| | EXP 400 | ja | 12 | 1, 2 | 30 | 4 | 65 | k. A. | a) 12,8x 39,8x31,6<br>b) nein | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | PO 200 | 1137,72 |
| | EXP 500 | ja | 16 | 1, 2 | 33 | 4 | 65 | k. A. | a) 12x44 x31<br>b) nein | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | PO 200 | 1365,72 |
| a) **Synelec Datensysteme**<br>b) Uchida | DWX-305 | nein | 20 | 1, 2 | 33 | 4 | ≤58 | 0,25 | a) 17x50 x35<br>b) nein | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | VC 64-Interf. 298,– | 1349,– (m. Centr.) |
| a) **Technitron GmbH**<br>b) Dyneer | DW 16 | nein | 16 | 1 | 30,5 | 3 | <65 | 96 Z | a) 12,2x 44,5x31,5<br>b) nein | 1, 2 | a) nein<br>b) nein<br>c) Aufpreis | Techni-tron | 1248,30 |
| | DW 20 | nein | 20 | 1 | 43,2 | 4 | <65 | 96 Z | a) 20x 58,3x38<br>b) nein | 1, 2 | a) nein<br>b) nein<br>c) Aufpreis | Techni-tron | 1818,30 |
| a) **Terhechte-Abels**<br>b) Juki | Juki 2200 | nein | 10 | 1 | 30 | 4 | k. A. | 2 | a) k. A.<br>b) ja | 1 (2, 3 opt.) | a) ja<br>b) ja<br>c) Aufpreis | Zett C 64 I/F | 998,– |
| a) **Triumph Adler**<br>b) Triumph Adler | Gabriele 8008 | nein | 10 | 1 | 34 | 4 | 68 | 0 | a) 13,4x 14,4x35,5<br>b) ja | 2, 3 | a) ja<br>b) ja<br>c) ja | k. A. | 998,– |
| | Gabriele 9009 | nein | 13 | 1 | 34 | 4 | 63 | 1,5 | a) 14,1x 42,1x40<br>b) ja | 1, 2, 3 | a) nein<br>b) nein<br>c) Aufpreis | IF-Box 564,– | 1058,– |

# Der erste Kontakt mit DFÜ

**Daten durch das Telefon zu schicken, ist einfacher als man denkt. Ein Neuling in Sachen DFÜ schildert seine Erfahrungen.**




**Zu solchen Leistungen (War Games) wird der Anfänger zwar noch nicht fähig sein, dennoch macht die Beschäftigung mit der DFÜ Spaß.**

Viele Mailboxen setzen Wissen voraus, das derjenige, der beabsichtigt, sich mit der Datenfernübertragung zu beschäftigen, in der Regel nicht hat. Hinzu kommen noch Anleitungen für Hard- und Software, die ebenfalls etwas Neues darstellen. Jeder kennt das Problem: Alles ist richtig angeschlossen, aber nichts funktioniert. Der Grund ist, daß neue Programme unzureichend bekannt sind.

## Der erste Kontakt mit einer Mailbox

Deshalb las ich mir die Anleitung zum Terminalprogramm genau durch und notierte mir die CTRL-Sequenzen, die für die Kommunikation mit einer Mailbox wichtig sind. Bevor das Terminalprogramm geladen werden konnte, mußte ich den C 64 hardwaremäßig für die Datenfernübertragung vorbereiten, indem ich ihn über ein RS232-Interface am User-Port mit dem Akustikkoppler verband, der laut Anleitung auf »originate« einzustellen ist. Es war mir nicht möglich, eine Erklärung für das Wort »originate« zu finden. In dem Handbuch zum Terminalprogramm (Terminal 64) stand nur folgendes:

»In der Betriebsart »Vollduplex« verwenden Sender und Empfänger unterschiedliche Bereiche (Originate/Answer). Für den Benutzer ist hier nur wichtig zu wissen, daß sein Modem jeweils entgegengesetzt der Einstellung seines Verbindungspartners sein muß.« Zitat Ende.

Nach dem Laden und Starten des Terminalprogramms erschien dessen Hauptmenü auf dem Bildschirm. Im Untermenü »Parameter einstellen« überzeugte ich mich davon, daß die Parameter mit denen des Handbuches übereinstimmten. Das heißt:

**Baudrate: 300 (300 Bit/Sekunde)**
**Anzahl der Daten-Bits: 7**
**Übertragungsmodus: Vollduplex**

Vollduplex bedeutet: Alle über die Tastatur eingegebenen Zeichen werden nicht unmittelbar, wie üblich auf dem Bildschirm dargestellt, sondern über den Akustikkoppler an die Mailbox gesendet, die dann ihrerseits jedes Zeichen wieder zurückschickt. Erst die zurückgeschickten Zeichen erscheinen auf dem Bildschirm. Dadurch kann der Sender kontrollieren, ob die gesendeten Zeichen richtig beim Empfänger angekommen sind.

**Parität: keine**

Das Paritäts-Bit entfällt. Wenn Parität auf gerade gestellt wird, werden alle Bits, die auf Eins stehen, addiert. Ist die Summe gerade, wird das Paritäts-Bit auf Null gesetzt, so daß die Quersumme des Datenwortes plus Paritäts-Bit immer eine gerade Zahl liefert.

Wird dem Empfänger eine ungerade Zahl angeboten, so schließt er daraus, daß die Übertragung fehlerhaft war.

**Stop-Bit: 1**

Jedem Datenwort wird ein Bit als Anhängsel angefügt, um Sender und Empfänger zu synchronisieren. Fehlt das Stop-Bit, so kann das Ende und der Anfang eines Datenwortes nicht erkannt werden.

Nachdem das Terminalprogramm geladen und die Parameter richtig eingestellt waren, stand dem ersten Datentransfer nichts mehr im Weg. Durch die Tastenkombination CTRL-← kam ich wieder zurück ins Hauptmenü und wählte durch das Untermenü »online« den Sende- und Empfangsbetrieb. Der Bildschirm wurde bis auf eine Informationszeile, in der Angaben über den momentanen Betriebszustand stehen, gelöscht. Weiter war nichts zu sehen. Auch der vertraute blinkende Cursor war verschwunden. Voller Zuversicht wählte ich die erste Mailboxnummer. Im Telefon war nur das Besetztzeichen zu hören. Ich wählte die nächste Mailboxnummer. Wieder war besetzt. Dieses wiederholte sich etwa 10 Minuten lang, bis ich bei einer Mailbox durchkam. Daß ich mit einer Mailbox verbunden war, erkannte ich daran, daß im Telefon ein langanhaltender Piepton zu hören war. Schnell legte ich den Hörer in die dafür vorgesehenen Gummimuffen des Akustikkopplers. Kaum war das geschehen, da rasten die ersten Zeichen mit einer Geschwindigkeit von 300 Baud über den Bildschirm. Nur mit Mühe konnte ich den an mir vorbeiflitzenden Text lesen, der Informationen

über die Benutzeranweisungen der Mailbox enthielt. Ich versuchte mir diese Informationen zu notieren, mußte jedoch schnell feststellen, daß es unmöglich ist, bei einer Übertragungsgeschwindigkeit von 300 Baud (Bit pro Sekunde) mitzuschreiben. Um die Benutzerhinweise nicht zu verpassen, machte ich von der Möglichkeit Gebrauch, den Datentransfer mit der Taste CTRL-S anzuhalten, notierte mir das, was mich interessierte und nahm den Datentransfer mit der Taste CTRL-Q wieder auf. Nachdem die Benutzeranweisungen an mir vorbeigerauscht waren, erwartete die Mailbox die Eingabe einer Benutzernummer, über die ich als Anfänger nicht verfügte. Ich versuchte es mit dem Wort »gast«. Als Gast war es mir im Gegensatz zu einem Anwender mit Benutzernummer, also einem eingetragenem Benutzer, nicht immer möglich, über die im System gespeicherten Programme zu verfügen. Auch ist der eingetragene Benutzer in der Lage Informationen abzurufen, die nur für ihn bestimmt sind. Ich tippte also das Wort »gast« ein. Kaum war es eingegeben, da ging die Post ab und das Hauptmenü der Mailbox erschien auf dem Bildschirm, was nicht immer der Fall ist. Häufig muß man, um ins Hauptmenü zu gelangen, RETURN, HELP, oder ein Fragezeichen eingeben. Ich wurde aufgefordert eines der zahlreichen Untermenüs anzuwählen und entschied mich für den Programm-Service, gab die entsprechende Zahl ein und sollte aus einigen aufgeführten Terminalprogrammen mein eigenes angeben. Leider war mein Programm nicht dabei. Da mich aber gerade der Programm-Service interessierte, gab ich ein anderes Terminalprogramm an. Der Erfolg war, daß mein Computer nichts mehr von sich gab. Ich versuchte die Verbindung ordnungsgemäß mit der Taste CTRL-X abzubrechen. Aber es passierte nichts. Auch der Versuch mit der Tastenkombination CTRL-←, mit der man bei vielen Terminalprogrammen zurück ins Hauptmenü gelangt, scheiterte. Offenbar war der C 64 abgestürzt. Ich war gezwungen, die Verbindung mit Gewalt abzubrechen, indem ich den Hörer auflegte, den Computer ausschaltete und das Terminalprogramm neu lud. Die weiteren Versuche, eine Mailbox anzurufen, verliefen mehr oder weniger erfolgreich.

## Lassen sich Mailboxen überlisten?

Hin und wieder kam es vor, daß Mailboxnummern, die aus einer Mailbox stammten, Privatanschlüsse waren. Deshalb sind diese Telefonnummern mit Vorsicht zu genießen. Es ist für die Betroffenen nun einmal sehr unangenehm, mitten in der Nacht aus dem Bett geklingelt zu werden.

## Lehrreiche Erfahrungen

Die Praxis hat gezeigt, daß sich Mailboxen nach dem ersten Freizeichen melden. Wenn die Leitung frei ist und die Mailbox meldet sich nicht nach dem ersten Freizeichen, handelt es sich entweder um eine Box die nur zeitweise betrieben wird oder man hat einen Privatanschluß vor sich.

Die meisten Terminalprogramme haben einen Modus, mit dem sich die empfangenen Daten auf Diskette abspeichern lassen. Leider verfügt nicht jede Mailbox über diese dem Anfänger sicherlich zu empfehlende Einrichtung.

Der Grund dafür ist, daß der C 64 die ankommenden Daten bis zu einer Länge von 5 KByte speichert und sie dann als Block auf Diskette schreibt. Während dieses Schreibvorgangs können keine Daten empfangen oder gesendet werden. Dieser Zustand wird der Mailbox über ein Stoppzeichen mitgeteilt. Einige Boxen können dieses Stopp-Zeichen nicht interpretieren und ignorieren es.

Daher kann es leicht passieren, daß der sogenannte Timeout erreicht wird, eine Zeitspanne, die bei den meisten Mailboxen zwischen 30 Sekunden und einer Minute liegt. Ist diese Zeitspanne überschritten, bricht die Mailbox die Verbindung ab, da sie annimmt, daß der Kommunikationspartner aufgelegt hat.

Ist eine bestimmte Mailbox besser bekannt, bietet sich eine Möglichkeit an, Gebühren zu sparen. Mit der Tastenkombination CTRL-C können bei einigen Mailboxen Systembeschreibungen und Menüunterpunkte übersprungen oder abgebrochen werden.

Auch beim Schreiben in die Mailbox (Briefkasten) kann man viel Zeit und Geld sparen, indem man nicht im Online-Modus (Sende-/Empfangsbetrieb) Texte eintippt, sondern den Editor des Terminalprogramms benutzt. Er gleicht von seiner Bedienung her einem Textverarbeitungsprogramm. Die fertigen Texte werden auf Diskette abgespeichert und können im Online-Modus direkt von der Diskette in die Mailbox geschrieben werden.

Jedes Terminalprogramm verfügt über die Möglichkeit, die Funktionstasten mit Texten zu belegen. Hier können zum Beispiel häufig benötigte Benutzernummern oder Adressen abgelegt werden.

Tauchen während einer Verbindung mit einer Mailbox Schwierigkeiten und Probleme auf oder werden Mängel an der Mailbox festgestellt, bieten die meisten Mailboxen ein Untermenü an, mit dem man sich direkt über den Computer mit dem Systemoperator in Verbindung setzen kann. Der Systemoperator ist derjenige, der für die Mailbox verantwortlich ist. Kann man ihn nicht direkt erreichen, besteht die Möglichkeit, ihm in einer Art Briefkasten Nachrichten zu hinterlassen. Auch Anfänger sollten nicht davor zurückschrecken, sich bei auftretenden Problemen an den Systemoperator zu wenden. Denn nur durch solche Nachrichten und Informationen kann man erreichen, daß die Mailboxen, die sich bis jetzt noch im Versuchsstadium befinden, oder für Anfänger (und auch für Profis) »Stolpersteine« darstellen, sprich über eine unklare Benutzerführung verfügen, verbessert und optimiert werden. Es gibt sogar Mailboxen, die erst nach der richtigen Beantwortung grundlegender Fragen zur Mailboxbenutzung den Zugang zur eigentlichen Mailbox erlauben.

## Datenfernübertragung ohne Mailbox

Jeder der über einen Computer mit RS232-Interface und Akustikkoppler verfügt, kann per Telefon mit anderen Computern ähnlicher Gerätekonfiguration über eine geeignete Treibersoftware (Terminalprogramme) kommunizieren. Dazu ist einer der Akustikkoppler auf »originate«, der andere auf »answer« einzustellen. Es spielt dabei keine Rolle, welcher von beiden auf »answer« und welcher auf »originate« steht. Der Akustikkoppler, der auf »answer« eingestellt ist, sendet nach Zustandekommen einer Verbindung einen Ton. Der Akustikkoppler, der auf »originate« eingestellt ist, antwortet mit einer Folge von Impulsen, die Informationen über die Parametereinstellungen enthalten. Dadurch ist zum Beispiel eine Mailbox in der Lage, sich an die Parametereinstellung (Baudrate, Parity, Stop-Bit etc.) eines Akustikkopplers anzupassen. Zwei miteinander kommuni-

zierende Computer können dies nicht, da Terminalprogramme die Folge von Impulsen, die Informationen über die Parametereinstellungen enthalten, nicht auswerten. Deshalb ist, wenn zwei Computer miteinander verbunden sind, besonders darauf zu achten, daß die Parameter übereinstimmen. Sonst kann es leicht passieren, daß plötzlich wirre, zusammenhanglose Zeichen auf dem Bildschirm erscheinen. Es ist dann aber nicht erforderlich, den Computer auszuschalten, oder die Verbindung abzubrechen. Denn während man sich im Online-Modus befindet, kann man bei vielen Terminalprogrammen mit der Tastenkombination CTRL-— ins Hauptmenü gelangen, von dort das Menü »Parametereinstellen« anwählen, die Parameter ändern und dann in umgekehrter Reihenfolge wieder in den Online-Modus gelangen. Wenn zwei Computer über das Telefon miteinander verbunden sind, bietet es sich natürlich auch an, Programme oder Listings auszutauschen. Um diesen Datentransfer zeitlich zu verkürzen, sollte versucht werden, den Austausch mit einer höheren Baudrate abzuwickeln. Bei 1200 Baud ist laut Aussage von Experten eine fehlerfreie Übertragung noch gewährleistet, vorausgesetzt, die Verbindung ist einwandfrei oder der Kommunikationspartner ist nicht allzuweit entfernt. Während man zum Beispiel für ein 30 KByte langes Programm bei 300 Baud etwa 18 Minuten benötigt, verkürzt sich die Übertragungszeit bei 1200 Baud auf etwa 4,5 Minuten.

Die Datenfernübertragung befindet sich heutzutage noch im Anfangsstadium. Sicherlich hat sie in Zukunft einen hohen gesellschaftlichen Stellenwert und auch ihren Nutzen sowohl im privaten als auch im kommerziellen Bereich. Aber schon heute ist der Entwicklungsstand der Mailboxen so weit fortgeschritten, daß es Spaß macht mit ihnen zu arbeiten. Jeder der über die entsprechende Soft- und Hardware verfügt, kann dort reichlich Informationen abrufen und das häufig 24 Stunden am Tag. Das Angebot der Mailboxen reicht von Mailboxzeitungen bis hin zu fertigen Programmen. In sogenannten Briefkästen können Nachrichten geschrieben und von jedem abgerufen werden. Für den Computerfreak stellen sie eine reiche Fundgrube dar. Hier wird alles angeboten, was in irgendeiner Form etwas mit dem Computer zu tun hat. Mich als Anfänger hat dieses Medium von Anfang an fasziniert. (ah)

# Terminalprogramme

**Wie überall braucht man auch in der Datenfernübertragung gute Werkzeuge. Ein solches Werkzeug ist für die DFÜ das Terminalprogramm.**

Das richtige Terminalprogramm auszuwählen, ist nicht so einfach, wie es auf den ersten Blick scheint. Für den Anwender ist es oft sehr schwer, die Spreu vom Weizen zu trennen. Eine Hilfe bieten bei der Entscheidung unsere Tests und unsere Marktübersicht. Nicht immer werden sich Fachausdrücke umgehen lassen. Für alle, die noch einmal genauer nachlesen wollen, haben wir auf Seite 44 ein kleines DFÜ-Lexikon abgedruckt.

Ein Terminalprogramm steuert die Kommunikation zwischen zwei Computern. Es ist in den meisten Fällen nicht von einem bestimmten Akustikkoppler oder Modem abhängig. Eine Ausnahme stellt zum Beispiel der »Ascom-Akustikkoppler« dar. Er wird nicht wie andere Koppler über den User-Port angeschlossen. Der Anschluß erfolgt über den Expansion-Port und erfordert deshalb eine andere Ansteuerung. Ein Terminalprogramm für dieses Gerät ist im Kaufpreis inbegriffen.

Gute Terminalprogramme bieten dem Benutzer die Möglichkeit, bestimmte Übertragungsparameter einzustellen. Der DFÜ-Neuling wird von diesen Einstellmöglichkeiten aber meist überfordert. Er sei an dieser Stelle auf den Artikel »DFÜ für Anfänger« hingewiesen.

Bevor wir jetzt zu den Tests kommen, noch ein wichtiger Hinweis: Achten Sie beim Kauf von Terminalprogrammen immer darauf, um welche Version es sich handelt! In der Vergangenheit sind einige dieser Programme innerhalb kurzer Zeit mehrmals verbessert oder erweitert worden. Dies wird von den Programmierern meist durch eine Erhöhung der »Versionsnummer« gekennzeichnet. Erkundigen Sie sich also möglichst vor dem Kauf bei Freunden oder beim Hersteller, welche Version die aktuellste ist. So sind Sie sicher, kein »veraltetes« Programm zu erwischen. Nun zu der Vorstellung zweier neuer Terminalprogramme.

Bei dem Programm »Dataterm 64« können Sie, wie bei den meisten Terminalprogrammen, zwischen deutscher und englischer Bedienerführung wählen. Nachdem die gewählte Sprache geladen ist, werden Sie aufgefordert, eine formatierte Datendiskette einzulegen. Diese wird für zwei Funktionen benötigt. Zum einen wird auf ihr, soweit man dies wünscht, der Dialog mit dem anderen Computer, zum anderen die Funktionstasten-Belegung abgespeichert.

## Dataterm 64

Die Grundeinstellung des Programms weist folgende Parameter auf: 300 Baud, 7 Datenbits, 1 Stop-Bit, 3-Draht-(Line-)Handshake und Vollduplex-Betrieb.

Der Pufferspeicher für übertragenen Text beträgt zirka 5000 Byte. Ist dieser gefüllt, wird zum Abspeichern automatisch ein »CTRL S« an die Gegenstelle gesendet, um den Datenfluß zu stoppen. Besitzt die Gegenstelle keine solche Stopp-Funktion, so wird dies dem Benutzer durch Dataterm 64 mitgeteilt. Nach dem Speichervorgang wird der Gegenstelle mit »CTRL Q« das Signal zum weiteren Senden gegeben. Während des Online-Betriebes kann der Speichervorgang mit der Pfeil-hoch-Taste abgebrochen und wieder aufgenommen werden.

Weitere Angaben zum Programm entnehmen Sie bitte unserer Marktübersicht.

Im Testbetrieb hat sich Dataterm 64 bewährt. Die Benutzerführung ist einfach zu bedienen und schnell erlernbar. Bei der Datenübertragung traten keine vom Programm bedingten Übertragungsfehler auf. Der Preis ist mit 49,90 Mark als äußerst günstig zu bezeichnen.

## Multicom 64 Version 1.3

Multicom 64 von Software Express gilt als der Nachfolger der Teleterm-Reihe. Es ist komplett in Maschinensprache geschrieben. Besonders erwähnenswert ist die ausführliche, gut verständliche Beschreibung, obwohl es sich derzeit noch um eine Vorversion handelt.

Die Parametereinstellung ist folgende: 300 Baud, 7 Datenbits, ein Stop-Bit, gerade Parität. Für den Fall, daß diese Parametereinstellung nicht die richtige ist (erkennbar an

unverständlichen Zeichen auf dem Bildschirm), bietet Multicom 64 eine Besonderheit. Mit dem Befehl »Auto-Logon« können Sie automatisch Übertragungsparameter ausprobieren. Das Programm sucht sich dann die passenden Parameter und stellt sie ein.

## Automatische Parametereinstellung

Das Haupt- und die Untermenüs bieten 47 einzelne Menüpunkte an. Zusätzlich gibt es noch sechs weitere Hilfsbefehle, die nicht in den Menüs aufgeführt werden. Unter diesen Befehlen befindet sich ein komfortabler Text-Editor, der fast schon einer Textverarbeitung gleicht. So verfügt er zum Beispiel auch über einen Zeilenumbruch. Das bedeutet, daß ein Wort, das nicht mehr in eine Zeile paßt, automatisch in die nächste Zeile geschoben wird.

Auch bei Multicom sind die Funktionstasten belegbar. Die Belegung kann auf der Datendiskette abgespeichert werden. Weiterhin verfügt das Programm über UPLOAD (Senden von Basic-Programmen) und DOWNLOAD (Empfangen von Basic-Programmen).

Zu erwähnen bleibt noch, daß der Benutzer sich von diesem Programm eine Telefonnummerndatei anlegen kann, die vom Programm aufrufbar ist. Der relativ hohe Preis von 198 Mark (inklusive Interface) für dieses Programm kann wegen seiner guten Leistungsmerkmale gerade noch als gerechtfertigt angesehen werden.

Negativ ist allerdings zu bemerken, daß dieses Programm nur zusammen mit der Software Expreß-Schnittstelle arbeitet. Wer also schon im Besitz einer RS232-Schnittstelle ist, sollte sich für ein anderes, mit ähnlichen Leistungsmerkmalen ausgestattetes Programm (zum Beispiel »Terminal 64«, Test 64'er Ausgabe 2/85) entscheiden. (rg)

**Kleine Marktübersicht der Terminalprogramme**

| Name | Baudrate | a) Datenbits b) Stop-Bits | Vollduplex Halbduplex | Parität ungerade gerade/ keine | Autoreturn | a) Tastatur Deutsch/ ASCII b) Drucker cbm/ASCII | a) Programmübertragung b) Funktionstastenbeleg c) Up-, Down-Load | Datenträger: Diskette Kassette Modul | Preis in Mark | Anbieter |
|---|---|---|---|---|---|---|---|---|---|---|
| Dataterm 64 | 10-1200 | a) 8/7/6/5 b) 1/2 | V/H | G/U/K | — | a) A b) C/A | a) J b) J c) U/D | D | 49,90 | HSV Streber |
| Procomm | 300 | a) 8/7/6/5 b) 1/2 | V/H | G/U/K | — | a) A b) C/A | a) J b) J c) U/D | D | 49,90 | HSV Streber |
| Tele AS 64 | 75-2400 | a) 8/7/6/5 b) 1/2 | V/H | G/U/K | aus | a) A b) C | a) J b) J c) — | D | 48 | Stockem Computertechnik |
| Terminal AS | 300 | a) 7 b) 1 | V | K | aus | a) A b) — | a) N b) N c) — | K/D | 8/9,90 | Stockem Computertechnik |
| Multicom V1.3 | 75-1200 | a) 8/7 b) 1/2 | V/H | G/U/K | 40 | a) A b) C/A | a) J b) J c) U/D | D/M | 198[1] | Software Express |
| Teleterm 3.0 | 50-1200 | a) 8/7 b) 1/2 | V/H | G/U/K | aus/40 | a) A b) C/A | a) J b) J c) U/D | D | 145[1] | Software Express |
| VIP Terminal | 50-1200 | a) 8/7 b) 1/2 | V/H | G/U/K | aus/40/ 64/80 | a) A b) C/A | a) J b) J c) U/D | D | 269 | Software Express |
| KOM 64 | 75-1200 | a) 8/7/6/5 b) 1/2 | V/H | G/U/K | aus/40 | a) A b) C/A | a) N b) J c) U/D | D | 118[1] [2] | CTK |
| Terminal 64 | 150-2400 | a) 8/7/6/5 b) 1/2 | V/H | G/U/K | aus/40/ 64/80/ 200 | a) D/A b) C/A | a) J b) J c) U/D | D | 169[1] | Elektronic Universe |
| Termulator | 75-1200 | a) 8/7/6/5 b) 1/2 | V/H | G/U/K | — | a) A b) — | a) J b) J c) — | D | 228 | Computer Related Products |
| Contakt 64 | 300 | a) 8/7 b) 1/2 | V/H | G/U/K | — | a) A b) — | a) J b) J c) — | D | [3] | Dynamics |
| Terminal und Telefon | 300 | a) 7 b) 2 | V | K | — | a) A b) C | a) N b) J c) U/D | D | 25 | Fotoelektronik Immo Drust |

1) Preis inklusive Interface 2) Mengenrabatt auch für Clubs 3) Gehört zum Lieferumfang des Ascom-Akustikkoppler. Nicht gesondert erhältlich (Gesamtpreis 249 Mark).

HSV-Streber
Pettenkoferstr. 24
8000 München 2
(089) 534903

Stockem Computertechnik
Berghausen 13
5778 Meschede
(0291) 1221

Software Express GmbH
Hugo-Viehoff-Straße 84
4000 Düsseldorf 30
(02 11) 424323)

CTK Computer-, Text- und Kommunikations-Systeme GmbH
Langenbrück 20
5060 Bergisch Gladbach 1
(02204) 63061

Electronic Universe A. Voigt-U. Wenk
Hindenburgstr. 98
2120 Lüneburg
(04131) 32760 + 37876

Computer Related Products
Claus F. Erbrecht
Lappenbergsallee 37
2000 Hamburg 20

Dynamics Marketing GmbH
Große Bäckerstr. 11
2000 Hamburg 1
(040)366147

Fotoelektronik Immo Drust
Landwehrstr. 5
6100 Darmstadt

# DFÜ — Was ist das?

**Modem**
Datenübertragungseinrichtung, die direkt an das Telefonnetz angeschlossen wird, also nicht über den Telefonapparat.

**Akustikkoppler**
Datenübertragungseinrichtung, die über das Telefon an das Postnetz angeschlossen wird. Erkennbar sind Akustikkoppler an den Gummimuffen für die Aufnahme des Telefonhörers.

**Handshaking**
»Handshaking« bezeichnet den Vorgang des Aussendens von Daten und des Wartens auf die Empfangsbestätigung des Computers, mit dem man den Datenaustausch vornimmt.

**Simplex**
Wenn auf einer Datenübertragungsleitung Daten nur in einer Richtung übertragen werden können, bezeichnet man dies als Simplexbetrieb. Dieses Verfahren wird wegen der fehlenden Rückmeldemöglichkeiten jedoch selten angewandt.

**Halbduplex**
Im Halbduplexbetrieb bestätigt der empfangende Computer den Empfang der Daten.

**Vollduplex**
Im Vollduplexbetrieb können beide Computer gleichzeitig senden. Weiterhin wiederholt der empfangende Computer die Daten und sendet sie an den ersten Computer zurück. Stimmen diese Daten überein, wird das nächste Byte gesendet. Sind sie nicht korrekt, werden sie wiederholt.

**Baud-Rate**
Die Baud-Rate bezeichnet die Geschwindigkeit mit der Daten übermittelt und empfangen werden. 300 Baud bedeutet, daß 300 Bits pro Sekunde übertragen werden.

**Paritätsbit**
Um die Übertragungssicherheit zu erhöhen, wird jedes Byte durch ein zusätzliches Bit ergänzt. Die Ergänzung erfolgt so, daß die Summe der gleichartigen Binärzustände (0 oder 1) eine ungerade oder gerade Zahl ergibt. Wird kein Bit ergänzt, so hat die Übertragung keine Parität.

**Start/Stop-Bits**
Einige Computer verlangen die Übertragung einer bestimmten Anzahl von Null-Bits nach jedem Byte. Die Benutzung von je einem Stop-Bit und einem Start-Bit ist am häufigsten.

**Treibersoftware, Terminalprogramm**
Mit Treibersoftware oder Terminalprogramm bezeichnet man die Programme, die die Datenübertragung steuern. Mit ihnen wird zum Beispiel die Übertragung von Stop-, Start- und Paritäts-Bits eingestellt und gesteuert.

**Mailbox**
Eine Mailbox ist eine Art elektronischer Briefkasten. Es wird zwischen drei Kategorien unterschieden:

1. Öffentliche Mailboxen (Zutritt für jedermann)
2. Private Mailboxen (Zutritt nur für eingetragene Benutzer)
3. Gemischte Mailboxen (öffentlicher und privater Bereich).

Alle drei Kategorien können gewerblichen Charakter haben. Im gewerblichen Bereich dienen Mailboxen oft zur Aufnahme von Bestellungen.

**Sysop**
Der System-Operator ist der Betreiber einer Mailbox oder die Aufsicht am Mailbox-Computer.

**Übertragungs-/Eingangsprotokoll**
Mit diesem Protokoll werden die Regeln festgelegt, die dem Übertragungsverfahren zugrunde liegen. Erst wenn beide Computer manuell oder automatisch auf das gleiche Übertragungsprotokoll eingestellt sind, kann eine sinnvolle Datenübertragung beginnen.

**Timeout**
Der Timeout ist die Zeitspanne, die eine Mailbox auf die Eingabe des Benutzers wartet, bevor sie annimmt, daß der Benutzer die Verbindung beendet hat. Die Telefonverbindung wird dann auch von der Mailbox beendet, damit die Leitung für den nächsten Anrufer zur Verfügung steht.

**Gesamtzeitlimit**
Das Gesamtzeitlimit ist das Zeitkonto des Benutzers innerhalb des Computersystems, mit dem er Kontakt aufgenommen hat. Erreicht der Anrufer sein Zeitlimit, wird die Verbindung automatisch beendet. Es soll verhindert werden, daß einzelne Benutzer einen Computer stundenlang blockieren. Das Zeitlimit von »eingetragenen Benutzern« ist meist größer bemessen, als das von »Besuchern«.

**Answer/Originate**
Wenn Sie mit einem Computer Verbindung aufnehmen, muß Ihr Modem oder Akustikkoppler auf »Originate« gesetzt sein. Dadurch wird das »Einführungsprotokoll« begonnen. Kommunizieren Sie direkt mit einem Computer, muß der eine im »Originate-«, der andere im »Answer-Modus« sein. Ist die Verbindung aufgebaut, spielt der Modus keine Rolle mehr.

**Datex-P**
Datex-P ist ein Verbindungsnetz der Bundespost. Über zentrale Vermittlungsstellen kann meist über große Entfernung zum Orts-Tarif kommuniziert werden. Siehe auch »Die Netze der Post«, Seite 46.

**Btx**
Service-Netz der Bundespost. Ausführlicher siehe auch »Die Netze der Post«, Seite 46.

**Telebox**
Die Telebox ist eine von der Bundespost betriebene Mailbox. Sie ist nur für eingetragene Benutzer konzipiert. Siehe auch »Die Netze der Post«, Seite 46.

**ISDN**
Mit ISDN (Integrated Services Digital Network) wird der bis 1988 geplante Netzausbau der Bundespost bezeichnet. Siehe auch »Die Netze der Post«, Seite 46.

**7-Bit-, 8-Bit-Übertragung**
Um den ASCII-Code darzustellen, benötigt der Computer nur sieben Bits. Mit dem Ausdruck 7-Bit, 8-Bit-Übertragung wird angegeben, ob in einem übertragenen Byte sieben Datenbits (also der ASCII-Wert) und ein »Füllbit« (7-Bit-Übertragung) oder sieben Datenbits und ein Paritätsbit (8-Bit-Übertragung) enthalten sind.

**Wortlänge**
Die Wortlänge bezeichnet die Anzahl der Bits in einem »Übertragungsbyte« beziehungsweise einer Übertragungseinheit. Sie besteht aus den Datenbits, dem Paritätsbit, dem Startbit und den Stop-Bit(s). Bei acht Datenbits, einem Paritätsbit, einem Startbit und zwei Stop-Bits ist die Wortlänge = 12. Aus »Schlampigkeit« wird in der DFÜ in diesem Fall oft von einem »Byte« gesprochen, obwohl es 12 Bits beinhaltet.

**Datenendeinrichtung**
Mit »Datenendeinrichtung« wird das Gerät »am Ende« einer Übertragungsleitung bezeichnet. Wird die Datenübertragung mit einem C 64 vorgenommen, so ist dieser die Datenendeinrichtung. Ein Modem oder Akustikkoppler ist dagegen ein Datenübertragungsgerät. (rg)

Sollten Ihnen Begriffe nicht klar geworden sein, oder sollten Sie Begriffe vermissen, die wir in dieser Übersicht nicht berücksichtigt haben, so schreiben Sie uns bitte.

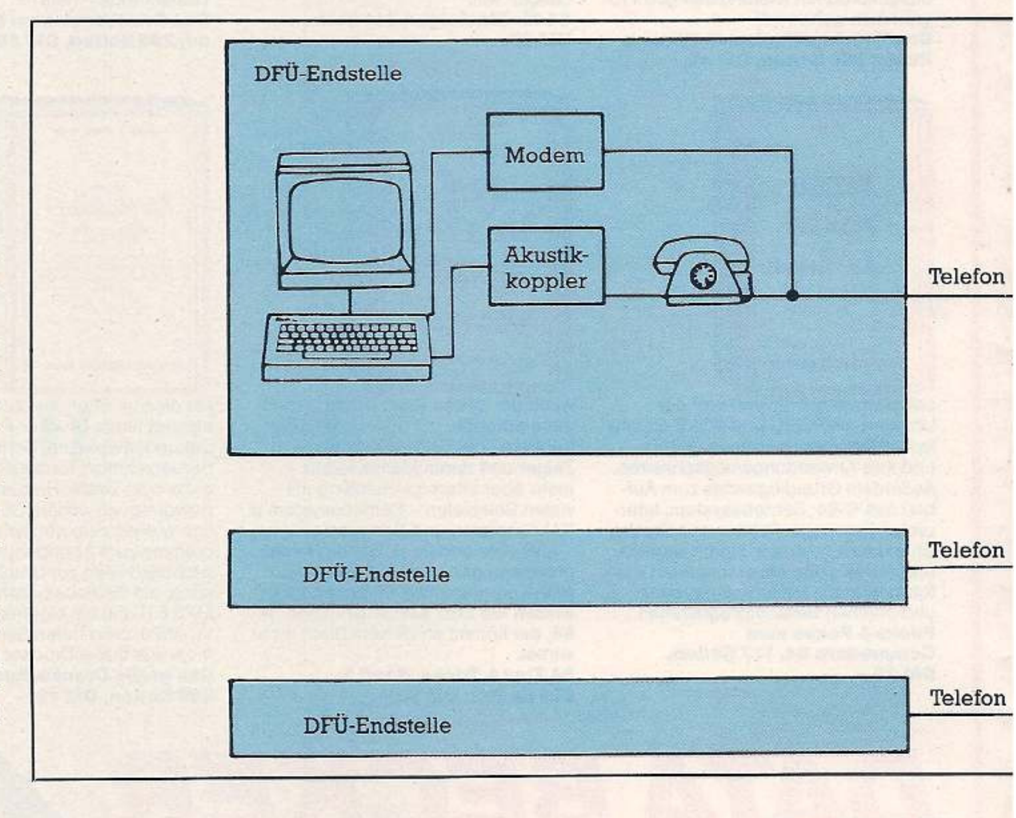

64ER ONLINE

# Die Netze der Post

**Die Datenfernübertragung mit Computern steckt in Deutschland eigentlich noch in den Kinderschuhen. Deshalb informieren wir Sie umfassend, was hinter den Begriffen Datex-P, Btx und Telebox steckt, und wie Sie diese Einrichtungen nutzen können. Auch die Route für die Zukunft hat die Bundespost schon festgelegt.**

Am 31. Dezember 1975 wurde von der Bundespost die FTZ-Prüfnummer 1813138000 an einen Teakholzkasten vergeben. Klappte man diesen Kasten auf, wurden zwei Mulden sichtbar, die einen Telefonhörer aufnehmen konnten. Dieser Kasten war ein Akustikkoppler. Genauer gesagt, der erste Akustikkoppler, der eine FTZ-Prüfnummer bekam.

Der Akustikkoppler im Holzkasten ist heute fast schon ein Museumsstück. In den letzten zehn Jahren haben ihn modernere und billigere Geräte abgelöst. Auch neue DFÜ-Netze sind mittlerweile von der Bundespost eingerichtet worden. So wurde 1980 das Datex-P-Netz eingeführt, 1983 Btx und 1984 die Telebox. Das Gesamtkonzept des weiterhin geplanten Netzausbaus bis 1988 wird mit ISDN bezeichnet. Was sich hinter den Begriffen Datex-P, Btx, Telebox und ISDN verbirgt, und wie man sich an diesen Netzen beteiligt, erfahren Sie im folgenden.



# Datex-P

Am 26. August 1980 begann im gesamten Bereich der Deutschen Bundespost ein zwölfmonatiger Probebetrieb für den Datex-P-Dienst. Direkt im Anschluß an den Probebetrieb wurde Datex-P dann in das feste Angebot der Bundespost übernommen. Doch was ist Datex-P eigentlich?

Datex-P ist ein öffentliches Datenübertragungsnetz, das binäre Daten (zum Beispiel von Computern; andere binäre Daten siehe ISDN) »in Paketen vermittelt«. Bei der Paketvermittlung werden die zu sendenden Daten zu Abschnitten (Paketen) zusammengefügt. Ein Paket kann bis zu 512 Bits enthalten. Ein Datenpaket wird gesendet, sobald es vollständig ist oder es nur noch den »Rest« einer Übertragung enthält. Es wird also nicht jedes Zeichen für sich übertragen, sondern die Zeichen werden zu kleinen Paketen gebündelt. Ist ein Datenpaket gesendet worden, wird die Übertragungsleitung für andere Benutzer freigegeben, bis das nächste Paket des gleichen Benutzers vollständig vorliegt. So wird eine Übertragungsleitung von mehreren Benutzern gleichzeitig verwendet. Man spricht hierbei von einer virtuellen Verbindung. Der Datex-P-Benutzer spürt von der virtuellen Verbindung jedoch überhaupt nichts. Er hat das Gefühl, daß er die Übertragungsleitung ständig zur Verfügung hat.

Berechnet wird jeweils nur die Datenmenge, die im Datex-P-Netz übertragen wird, und die Zeit, während der sich ein Benutzer in das Netz einschaltet. Die Entfernung spielt keine Rolle. So ist es zum Beispiel egal, ob Sie Daten von München nach Nürnberg oder von München nach Hamburg übertragen.

## Bedienung

Um Datex-P zu nutzen, brauchen Sie nur einen Computer, einen Akustikkoppler oder ein Modem und ein Terminalprogramm. Über das Telefon wird dann die nächste Datex-P-Vermittlungsstelle angewählt (Rufnummern siehe Tabelle 1.). Ist die Verbindung hergestellt (der Akustikkoppler oder das Modem schaltet auf READY), muß ein ».« (Punkt) und RETURN eingegeben werden (Dienstanforderungssignal). Die Vermittlungsstelle meldet sich zur Kontrolle dann mit: »Datex-P: 44 xxxx xxxxx«. Hierbei bezeichnet 44 xxxx xxxxx die Datex-P-Rufnummer des erreichten Anschlusses in der Vermittlungsstelle. In München könnte diese Meldung zum Beispiel »Datex-P: 44 8900 49130« lauten.

Die Vermittlungsstelle erwartet jetzt die Eingabe einer NUI (**N**etwork **U**ser **I**dentification) oder NUA (**N**etwork **U**ser **A**dress). Die NUI ist ein Code, der dem System mitteilt, daß Sie die Gebühren der Datenübertragung übernehmen. Die NUI muß bei der Bundespost beantragt werden. Die Eingabe würde lauten: NUI Dxxxxx, wobei xxxxx die Teilnehmerkennung ist. Nach dieser Eingabe wird noch Ihr Passwort abgefragt.

Die NUA ist die Datex-P-Rufnummer des Teilnehmers, den Sie anwählen wollen. Besitzen Sie keine NUI, muß die Gegenstelle bereit sein, die Gebühren zu übernehmen. Hierzu müssen Sie der Rufnummer (NUA) ein »R« (Reverse Charging) voranstellen. Die Datex-P-Vermittlungsstelle prüft dann, ob die Gegenstelle mit der Gebührenübernahme einverstanden ist. Die NUA kann wie folgt aussehen (dies ist eine Fantasienummer):

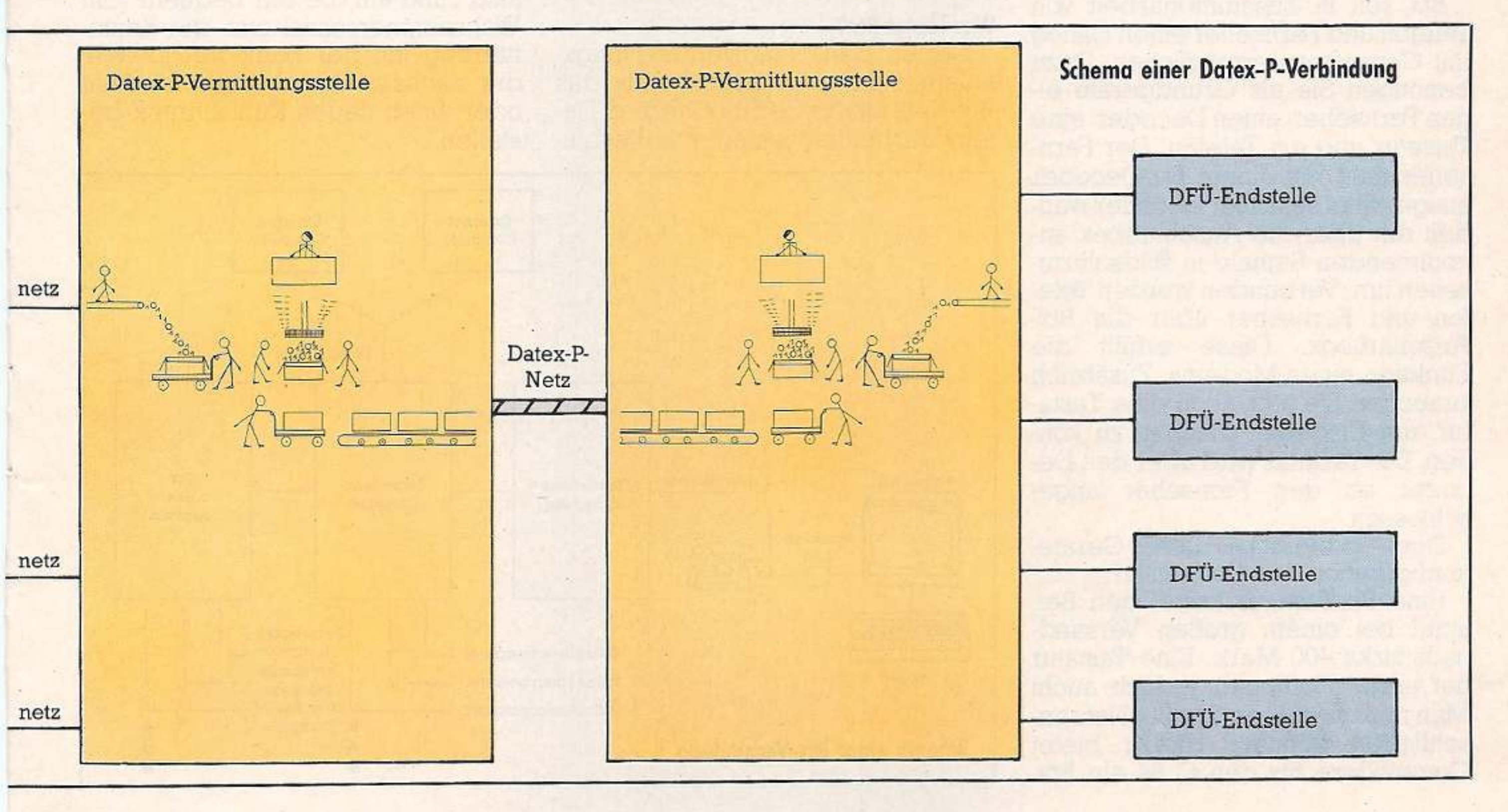
45 8900 12345 (Rufnummer)
R 45 8900 12345 (Rufnummer mit der Bitte um Gebührenübernahme)
C(xxx) 45 8900 12345 (Rufnummer mit Teilnehmerklasse, durch den Index xxx gekennzeichnet. Die Teilnehmerklasse kennzeichnet die Bereiche einer Datenbank, die benutzt werden können).

## Gebühren

Bei Datex-P gibt es, abweichend vom Telefontarif, drei Zeitabschnitte mit unterschiedlichen Verbindungsgebühren (Tabelle 2):

Tag — von 8 bis 18 Uhr
Nacht 1 — von 6 bis 8 Uhr und von 18 bis 22 Uhr
Nacht 2 — von 22 bis 6 Uhr und Samstag 14 Uhr bis Montag 6 Uhr, sowie an bundeseinheitlichen Feiertagen von 0 bis 24 Uhr.

Die Gebühren setzen sich aus verschiedenen Einzelpositionen zusammen, die Sie dem »Bundespost



**Schema einer Datex-P-Verbindung**

Informationsblatt Nummer 146« entnehmen können. An dieser Stelle seien nur zwei kleine Beispiele aus dem »Informationsblatt Nummer 145« der Bundespost angeführt:

Für eine Verbindung mit einer Datenbank von 30 Minuten Dauer bei zirka 7500 übertragenen Zeichen ist eine Gebühr von rund 0,75 Mark an die Post zu zahlen.

Bei einer Reisebuchung im Dialog einschließlich der Ausgabe der Reiseunterlagen von zwei Minuten Dauer mit zirka 2500 übertragenen Zeichen fällt eine Gebühr von etwa 0,20 Mark an.

In beiden Fällen kann diese Gebühr unter Berücksichtigung der mengenabhängigen Gebührenstaffel und der Tageszeit noch geringer ausfallen (siehe Tabelle 2). Eine NUI kostet 15 Mark im Monat.

Anträge für einen Datex-P-Hauptanschluß und die Erteilung einer NUI sind bei Ihrem örtlichen Fernmeldeamt erhältlich.

Wer sich noch weiter über Datex-P informieren will, kann bei der Bundespost die Informationsblätter Nummer 93, 145, 146, 166 und 189 bestellen.

| DATEX-P Vermittlungsstelle | Ortsnetzkennzahl (Vorwahl) | Ruf-Nr. für die Übertragungsgeschwindigkeiten bis 300 Bit/s (D300S) | 1200 Bit/s (D1200S12) | 1200/75 Bit/s (D1200S0) |
|---|---|---|---|---|
| Augsburg | 0821 | 464011 ( 36791) | 464031 ( 36781) | 464051 ( 36761) |
| Berlin | 030 | 240001 | 240081 | 240061 |
| Bielefeld | 0521 | 59011 | 59021 | 59041 |
| Bremen | 0421 | 170131 | 14291 | 15077 |
| Dortmund | 0231 | 57011 | 52011 | 52081 |
| Düsseldorf | 0211 | 329318 | 329249 | 320748 |
| Essen | 0201 | 787051 | 791021 | 793003 |
| Frankfurt | | 20281 | 20291 | 20201 |
| Hamburg | 040 | 441231 | 441261 | 441281 |
| Hannover | 0511 | 326651 | 327481 | 327591 |
| Karlsruhe | 0721 | 60241 | 60381 | 60581 |
| Köln | 0221 | 2911 | 2931 | 2951 |
| Mannheim | 0621 | 409085 | 39941 | 39951 |
| München | 089 | 228730 | 228630 | 228758 |
| Nürnberg | 0911 | 20571 | 20541 | 20501 |
| Saarbrücken | 0681 | 810011 | 810031 | 810061 |
| Stuttgart | 0711 | 299171 | 299061 | 299291 |

**Tabelle 1. Rufnummern der Datex-P-Vermittlungsstellen**

| Zeitabschnitte / Mengenstaffel | Volumengebühr (national) Pf/Segment Taggebühr 08.00-18.00 | Nachtgebühr I 6-8 und 18-22 | Nachtgebühr II 22.00-06.00 |
|---|---|---|---|
| 1. 0,2 Mio Segmente | 0,33 | 0,18 | 0,09 |
| 2. 0,2 Mio Segmente | 0,18 | 0,12 | 0,06 |
| alle weiteren Segm. | 0,12 | 0,08 | 0,04 |

**Tabelle 2. Die Datex-P-Verbindungsgebühren**

# Bildschirmtext

Bildschirmtext, kurz Btx, wurde von der Deutschen Bundespost im September 1983 als allgemeiner Fernmeldedienst eingeführt. Allerdings lief Btx bis zur Inbetriebnahme einer neuen Systemtechnik Mitte 1984 nur mit Einschränkungen.

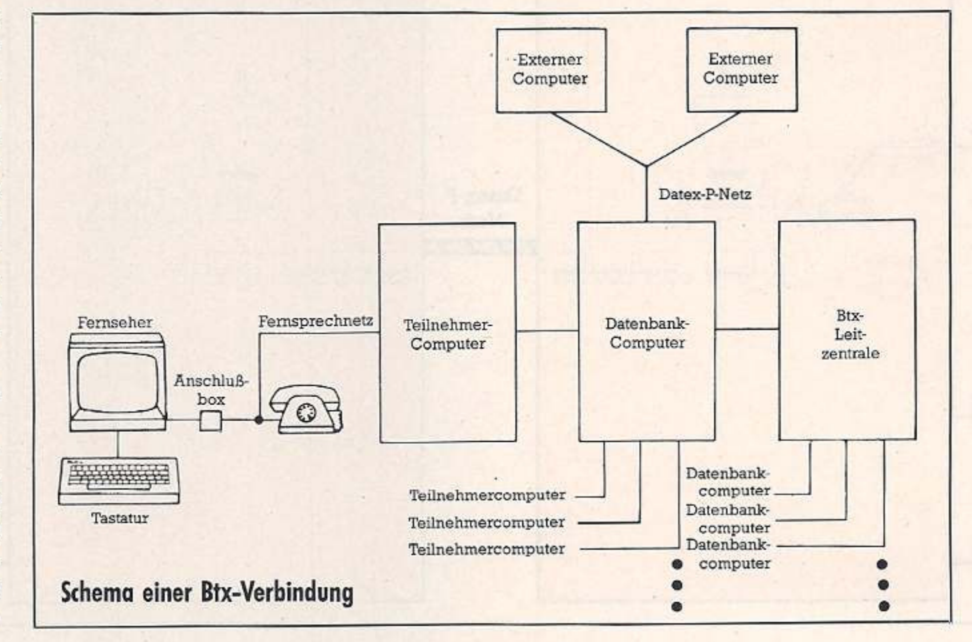
Btx soll in Zusammenarbeit von Telefon und Fernseher einen Dialog mit Computern ermöglichen. Dazu benötigen Sie als Grundgeräte einen Fernseher, einen Decoder, eine Tastatur und ein Telefon. Der Fernseher muß mit einem Btx-Decoder ausgestattet sein. Der Decoder wandelt die über die Anschlußbox ankommenden Signale in Bildschirmseiten um. Verbunden werden Telefon und Fernseher über die Btx-Anschlußbox. Diese erfüllt die Funktion eines Modems. Zusätzlich brauchen Sie jetzt noch eine Tastatur, um Eingaben machen zu können. Die Tastatur wird über den Decoder an den Fernseher angeschlossen.

Doch wo bleibt bei dieser Gerätekonfiguration der Computer?

Eine Btx-Tastatur kostet zum Beispiel bei einem großen Versandhaus zirka 400 Mark. Eine Tastatur hat unser Computer jedoch auch! Man muß sie nur an das Btx-Netz anschließen können. Hierzu bietet Commodore für den C 64 ein Btx-Modul an, das diesen Anschluß ermöglicht. Zusätzlich erlaubt das Modul das Speichern und Laden von Bildschirmseiten, was bei Btx sonst nicht möglich wäre. Bei einem Preis von 298 Mark für das Modul spart man also 100 Mark.

### Was leistet Btx?

Neben den Datenverarbeitungsanlagen der Bundespost steht das Btx-Netz mit zur Zeit 89 externen Datenverarbeitungsanlagen in Verbindung (übrigens über Datex-P). Zu diesen gehören zum Beispiel Computer von Versandhäusern, Banken, Reiseveranstaltern etc. Der Benutzer kann unmittelbar mit Bestell-, Buchungs- und Reservierungssystemen Kontakt aufnehmen. Man kann also, rund um die Uhr bequem vom Wohnzimmersessel aus, die Kontoführung mit der Bank bearbeiten, die nächste Urlaubsreise buchen oder einen neuen Kühlschrank bestellen.



**Schema einer Btx-Verbindung**




**Bild 1. Mit dem neuen CEPT-Standard können Bildschirmseiten in hochauflösender Grafik dargestellt werden. Hier ein Bild der österreichischen Fremdenverkehrswerbung.**




**Bild 2. So sahen die Bildschirmseiten vor der Änderung des Standards aus**

Über Btx können umfangreiche Datenbanken abgerufen und von Rechenzentren angebotene Datenverarbeitungsprogramme für Berechnungen genutzt werden.

Es gibt unter den Anbietern von Bildschirmseiten auch Zeitungsverlage, die zum Beispiel die neuesten Nachrichten, Lokalberichte oder den Wetterbericht anbieten.

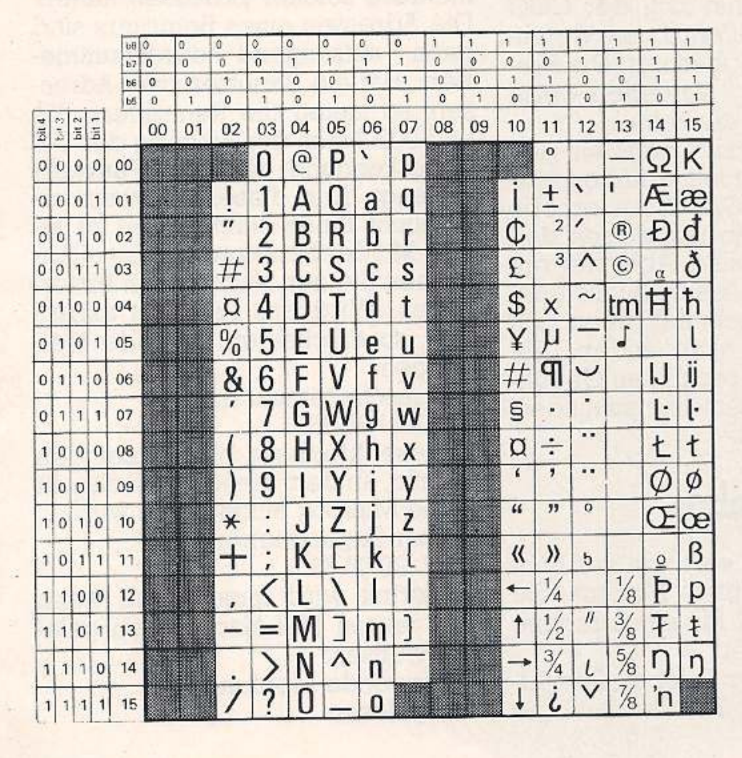
Und noch eine weitere Nutzungsmöglichkeit wird zur Zeit in den Bildschirmtext integriert: Die Telesoftware. Unter diesem Namen wird in Zukunft Software für Home- und Personal Computer abrufbar sein. (Siehe auch unseren »Btx-Wettbewerb« in dieser Ausgabe.) Die Gebühren für diesen neuen Service stehen zur Zeit noch nicht fest.

**Btx-Übertragungsstandard**

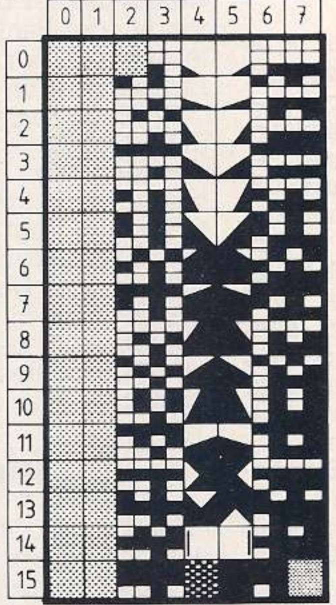
Seit September 1983 gilt für Bildschirmtext ein neuer, internationaler Darstellungsstandard, der von der CEPT (Konferenz der Europäischen Verwaltungen für Post und Fernmeldewesen) festgelegt wurde. Er umfaßt ein Basisrepertoire von 335 lateinischen Buchstaben, Ziffern und Zeichen. Zusätzlich erlaubt der CEPT-Standard bis zu 94 frei gestaltbare Zeichen in den Decoder zu laden. Dieser Punkt macht es zum Beispiel erst möglich, Programme mit den Grafikzeichen des C 64 zu übertragen, da diese nicht im bisherigen Basisrepertoire von Btx enthalten sind. Dieser frei definierbare Zeichensatz wird mit DRCS (Dynamically redifinable character set) bezeichnet.

Grafiken werden aus »Mosaikfeldern« zusammengesetzt. Mosaikfelder bestehen aus einem zwei mal drei Felder großen Raster. Die Mosaikelemente werden in dieses Raster eingesetzt, wobei insgesamt 64 Kombinationsmöglichkeiten zur Verfügung stehen. Dieses Darstellungsverfahren wird allgemein als das »Alphamosaikverfahren« bezeichnet. Zusätzlich enthält der neue Standard 64 Schrägflächen- und 32 Linienzeichen (Tabelle 3).

Den Unterschied zwischen dem alten und dem neuen Standard können Sie in Bild 1 und Bild 2 erkennen. Bild 1 verdeutlicht anschaulich die



| | | | | b8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | b7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | | | b6 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | | | | b5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| bit 4 | bit 3 | bit 2 | bit 1 | | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 | 0 | 0 | 00 | | | | 0 | @ | P | ` | p | | | | ° | | — | Ω | K |
| 0 | 0 | 0 | 1 | 01 | | | ! | 1 | A | Q | a | q | | | ¡ | ± | ` | ' | Æ | æ |
| 0 | 0 | 1 | 0 | 02 | | | " | 2 | B | R | b | r | | | ¢ | ² | ´ | ® | Đ | đ |
| 0 | 0 | 1 | 1 | 03 | | | # | 3 | C | S | c | s | | | £ | ³ | ^ | © | ª | ð |
| 0 | 1 | 0 | 0 | 04 | | | ¤ | 4 | D | T | d | t | | | $ | × | ~ | tm | Ħ | ħ |
| 0 | 1 | 0 | 1 | 05 | | | % | 5 | E | U | e | u | | | ¥ | µ | ¯ | ♪ | | ı |
| 0 | 1 | 1 | 0 | 06 | | | & | 6 | F | V | f | v | | | # | ¶ | ˘ | | Ĳ | ĳ |
| 0 | 1 | 1 | 1 | 07 | | | ' | 7 | G | W | g | w | | | § | · | ˙ | | Ŀ | ŀ |
| 1 | 0 | 0 | 0 | 08 | | | ( | 8 | H | X | h | x | | | ¤ | ÷ | ¨ | | Ł | ł |
| 1 | 0 | 0 | 1 | 09 | | | ) | 9 | I | Y | i | y | | | ‘ | ’ | ¨ | | Ø | ø |
| 1 | 0 | 1 | 0 | 10 | | | * | : | J | Z | j | z | | | “ | ” | ˚ | | Œ | œ |
| 1 | 0 | 1 | 1 | 11 | | | + | ; | K | [ | k | { | | | « | » | ¸ | | º | ß |
| 1 | 1 | 0 | 0 | 12 | | | , | < | L | \ | l | \| | | | ← | ¼ | | ⅛ | Þ | þ |
| 1 | 1 | 0 | 1 | 13 | | | – | = | M | ] | m | } | | | ↑ | ½ | ˝ | ⅜ | Ŧ | ŧ |
| 1 | 1 | 1 | 0 | 14 | | | . | > | N | ^ | n | ‾ | | | → | ¾ | ˛ | ⅝ | Ŋ | ŋ |
| 1 | 1 | 1 | 1 | 15 | | | / | ? | O | _ | o | | | | ↓ | ¿ | ˇ | ⅞ | ŉ | |



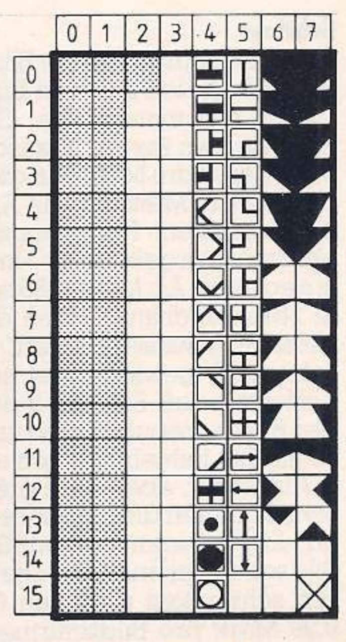


**Tabelle 3. Der Basis-Zeichensatz, die Mosaikelemente und die Schräg- und Linienflächen von Bildschirmtext**

hochauflösende Grafik des neuen Standards. Bild 2 zeigt den der Blockgrafik des C 64 entsprechenden alten Standard.

**Wie funktioniert Btx ?**

Wählt man das Btx-Netz über Telefon an, wird man mit einem »Teilnehmerrechner« verbunden. Dieser kann jeweils bis zu 100 Btx-Verbindungen gleichzeitig bearbeiten. Im Teilnehmerrechner sind über 50000 Bildschirmseiten gespeichert. Steht die von Ihnen gewünschte Bildschirmseite hier nicht zur Verfügung, wird die Anfrage an den dem Teilnehmerrechner übergeordneten Datenbankrechner weitergeleitet. In diesem sind 90000 Bildschirmseiten gespeichert. Erst wenn auch hier die gewünschte Bildschirmseite nicht vorliegt, wird eine Verbindung mit der Btx-Leitzentrale in Ulm hergestellt. Dies soll nach Aussage der Bundespost jedoch nur in zwei Prozent aller Fälle notwendig sein.

Die Auswahl der im Teilnehmer- und Datenbankrechner angebotenen Bildschirmseiten richtet sich nach der Häufigkeit der diese Seiten betreffenden Aufrufe. Wird zum Beispiel eine im Teilnehmerrechner gespeicherte Seite neunmal verlangt, eine im Datenbankrechner gespeicherte aber 40mal, so werden diese ausgetauscht. Das heißt die wenig verlangte Bildschirmseite wird im Datenbankrechner gespeichert, die häufig verlangte im Teilnehmerrechner. So ist immer ein schneller Zugriff auf die meist verlangten Bildschirmseiten gewährleistet.

**Gebühren**

Die Gebühren für den Bildschirmtext setzen sich aus drei Einzelpositionen zusammen. Die Grundgebühr für den Btx-Hauptanschluß kostet 8 Mark pro Monat. In dieser Gebühr ist die Miete für die Anschlußbox enthalten. Hinzu kommen die Verbindungsgebühren im Fernsprechnetz. Ab Mitte 1985 soll für alle Fernsprechanschlüsse der Ortsbeziehungsweise Nahtarif zu Bildschirmtext gewährleistet sein. Das bedeutet, daß Sie von jeder Stelle der Bundesrepublik aus zum Ortstarif am Btx teilnehmen können.

Als letzter, aber auch größter Posten, kommen die Gebühren für die im allgemeinen kostenpflichtigen Bildschirmseiten dazu. Die Gebühren schwanken zwischen 0,00 und 9,99 Mark pro Bildschirmseite. Die Gebührenhöhe bleibt in diesem Spielraum dem Anbieter der Bildschirmseite überlassen. Auf diese Gebühren muß vom Anbieter bei einem Aufruf hingewiesen werden. So kann der Btx-Benutzer immer noch entscheiden, ob er diese Gebühren akzeptiert.

Alle, die sich weiter über Bildschirmtext informieren wollen, können den Sonderdruck aus dem Dokumentarband Bundespost '84 »Die Technik des Bildschirmtextdienst« bei der Bundespost bestellen.

**Ist Btx nicht sicher?**

Ende 1984 machte eine Nachricht in der Presse Schlagzeilen. Der Hamburger Chaos Computer Club (CCC) hatte nach eigenen Angaben das Btx-Sicherheitsnetz umgangen und einen »Bankraub« durchgeführt. Was genau war passiert? Der CCC hatte einen Softwarefehler der Btx-Steuerung entdeckt und sich das Passwort der Hamburger Sparkasse beschafft. Erst mittels dieser Kombination war es dem CCC möglich, durch ein Programm, das auf einem Personal Computer lief, eine gebührenpflichtige Bildschirmseite über 10000mal innerhalb einer Nacht abzurufen. Da das Passwort der Bank verwendet wurde, wurden dieser die Gebühren berechnet. In diesem Fall zirka 135000 Mark.

Dazu die Bundespost: »Wie der CCC bei dieser Aktion im einzelnen vorging, ist bis heute noch nicht geklärt. Der Softwarefehler lag tatsächlich vor, ist heute aber behoben. Dieser Fehler konnte aber nur im Zusammenhang mit einem Passwort ausgenutzt werden. Das Passwort der Hamburger Sparkasse, so nimmt man an, hat sich der CCC wohl bei einer öffentlichen Vorführung in der Bank besorgt. Die Hintergründe der Vorgehensweise bleiben aber im dunkeln«.

Es bleibt noch zu erwähnen, daß diese Gebühren nicht erhoben worden sind. Wer Gebühren erhebt, muß auf Verlangen nachweisen, daß diese berechtigt sind. Aber wer ruft über 10000mal dieselbe Bildschirmseite in einer Nacht auf, wenn sich die Information nicht ändert. Ein Nachweis der berechtigten Gebühren wäre in diesem Fall zumindest fraglich gewesen.

## Telebox

Die Telebox ist eine von der Bundespost eingerichtete Mailbox. Sie ist ausschließlich für eingetragene Benutzer konzipiert.

Installiert ist der Telebox-Computer in Mannheim. Zur Zeit läuft die Telebox noch in einem zweistufigen Testbetrieb. Die erste Stufe begann Juni 1984 und endete am 31.09.84. Unterbrechungsfrei ging sie am 01.10.84 in die zweite Stufe über. Der Testbetrieb wird am 30.09.85 abgeschlossen sein.

**Benutzung**

Der Telebox-Computer ist unter folgenden Rufnummern erreichbar:
Telefonnetz: 0621/413091, 300 Baud
0621/412071, 1200 Baud
Datex-P (NUI erforderlich):
45 6210 40000
45 6210 90000.

Wird die Telebox angewählt, erhalten Sie die Aufforderung »Bitte vorstellen«. Wer sich nur über das Telebox-System informieren will, beantwortet diese Aufforderung mit: ID INF100 TELEBOX

Sie können dann auf umfangreiche Informationsdateien zugreifen, die Sie über das Leistungsangebot der Telebox unterrichten.

Eingetragene Benutzer oder Anrufer, die einem Benutzer eine Nachricht hinterlassen wollen, melden sich mit einer Benutzerkennung. Diese ist mit einer Telefonnummer im Fernsprechnetz vergleichbar. Ein Verzeichnis der Benutzernummern finden Sie in den Informationsdateien. Die Benutzernummer wird wie folgt eingegeben:
ID ABC123

ABC ist das Kürzel des eingetragenen Benutzers. 123 gibt die Nummer des Speicherbereiches (Adresse) an, in dem eine Nachricht gelesen oder geschrieben werden soll. Ein eingetragener Benutzer kann mehrere solcher Adressen haben. Die Adressen eines Benutzers sind dann aufsteigend durchnummeriert. Hat ein Benutzer zwei Adressen, so lauten die Kennziffern 001 und 002. Nach der Eingabe der Benutzerkennung wird ein Passwort verlangt. Erst dieses Passwort ermöglicht Ihnen Nachrichten zu lesen. Mit der Benutzeradresse allein können Sie nur Nachrichten hinterlassen. Wird das Passwort akzeptiert, so erscheinen folgende Meldungen:

Ausgeschaltet um/am:
12.00/21.06.85

Diese Meldung bezeichnet den Termin, an dem die letzte Nachricht in die Mailbox eingegeben wurde.

Letzter Zugang um/am:
17.01/20.06.85

Hiermit wird angegeben, wann zum letzten Mal Nachrichten gelesen wurden.

Vorliegende Meldungen:
3 ungelesen

Diese Meldung besagt, daß drei Nachrichten bisher vom Benutzer noch nicht gelesen wurden. Hierin liegt ein großer Vorteil des Telebox-Systems, denn Sie können gleich an der Eingangsmeldung erkennen, ob neue Nachrichten für Sie vorliegen. Sie müssen also nicht erst weiter in das System vordringen.

Will man das System verlassen, so muß »ENDE« eingegeben werden.

**Gebühren**

Ab 01.10.84 kostet jede Adresse 40 Mark Grundgebühr plus 40 Mark pauschale Nutzungsgebühr.

Nach Ende der Testphase am 30.09.85 werden folgende Gebühren fällig:

— Belegungsgebühr je Minute 0,30 Mark

— Speichergebühr je Tag und Einheit (2 KByte) 0,03 Mark

— Adressiergebühr je Adresse 0,10 Mark

Mindestnutzungsgebühr sind 40 Mark pro Monat.

Um die Gebühren der Telebox zu erläutern, und um sie dem Briefverkehr gegenüberzustellen, ein kleines Beispiel:

Ein Verkäufer einer Hamburger Firma hat in Frankfurt einen Verkaufsabschluß getätigt. Er möchte den Text und die Daten an seine Firma übermitteln. Diese hat eine Telebox-Adresse. Daten und Text umfassen 10 KByte; der Verkäufer benutzt einen 300 Baud Akustikkoppler. Der Text und die Daten sind vorbereitet und werden gesammelt übertragen.

| | |
|---|---|
| Gebühr für Ortsgespräch | 0,23 Mark |
| Datex-P-Gebühr | 0,33 Mark |
| Belegungsgebühr Telebox | 0,30 Mark |
| Speichergebühr (1 Tag) | 0,15 Mark |
| Adressierungsgebühr | 0,10 Mark |
| Übermittlung Verkäufer zur Firma | = 1,11 Mark |
| Gebühr für Ortsgespräch | 0,23 Mark |
| Datex-P-Gebühr | 0,33 Mark |
| Abruf der Daten | 0,56 Mark |
| Anfallende Kosten für die Firma gesamt: | 1,67 Mark |

Die Daten sind für die Firma sofort verfügbar.

Stellen wir jetzt die Übermittlung per Brief gegenüber:

| | |
|---|---|
| Porto | 0,80 Mark |
| Einschreibgebühren | 2 Mark |
| Eilbrief | 3 Mark |
| Gesamt: | 5,80 Mark |

Die Firma muß je nach Uhrzeit und Verbindungsweg (Flugzeug oder Bundesbahn) drei Stunden oder bis zum nächsten Morgen warten.

Wenn Sie eine eingehende Beratung über das Teleboxsystem wünschen, wenden Sie sich bitte an folgende Adresse:

Fernmeldetechnisches Zentralamt
Referat T 21
Postfach 5000
6100 Darmstadt
Tel. 06151/83-4641

Den Antrag für die Teilnahme am Telebox-Betrieb erhalten Sie beim:

Fernmeldeamt
Anmeldestelle für Fernmeldeeinrichtungen
Postfach 7300
6800 Mannheim 1
Tel. 0621/24853

# ISDN — Digitalisierte Zukunft

»ISDN — die Antwort der Deutschen Bundespost auf die Anforderungen der Telekommunikation von Morgen«, so wird von der Bundespost das Konzept für die Zukunft bezeichnet.

ISDN (Integrated Services Digital Network) heißt also unsere »fernmeldetechnische Zukunft«. Die kurze Umschreibung der Bundespost sagt nicht gerade viel aus. Äußerungen wie »Die Deutsche Bundespost wird künftig auch Fernmeldedienstleistungen über Satelliten anbieten, mit denen die terrestrischen Netze schnell und flächendeckend ergänzt werden« scheinen einem Science Fiction-Roman entnommen. Doch so steht es in einer Informationsbroschüre der Deutschen Bundespost.

Was hat die Bundespost nun wirklich geplant? Uns allen dürfte klar sein, daß die Technik in vielen Teilen des Fernmeldedienstes überaltert ist. Noch heute ist das Rattern der Relais in den Vermittlungsstellen der Fernmeldeämter zu hören, wenn eine Telefonverbindung hergestellt wird.

1976 wurde mit dem Aufbau des IDN (Integriertes Text- und Datennetz) begonnen. Zum IDN gehören auch Telex und Datex-P. Zur Zeit sind an das IDN über 300000 Teilnehmer angeschlossen.

Es handelt sich hierbei um ein Fernmeldenetz, in dem die digitale Übertragungs- und Vermittlungstechnik angewandt wird.

**Das Digitale Fernsprechnetz**

Was bedeutet »Digitalisierung«? Nehmen wir als Beispiel ein Telefongespräch. Wie sieht das bisher aus? Man wählt eine Rufnummer. Die Verbindung wird dann durch einzelne Relais in der Vermittlungszentrale geschaltet. Dies dauert seine Zeit. Ist die Verbindung hergestellt, wird die Sprache, die das Mikrofon im Telefonhörer aufnimmt, in elektromagnetische Signale umgesetzt. Diese werden dann übertragen und müssen, da sie bei der Übertragung über die jetzigen Leitungen der Post schnell an Intensität abnehmen, oft verstärkt werden. Die ankommenden Schwingungen werden vom Lautsprecher des Telefonhörers wieder in Töne verwandelt.

Im digitalisierten Fernsprechnetz funktioniert dies anders.

Alle Signale werden im Binär-Code übertragen. Als erstes erreicht die codierte Rufnummer den Vermittlungscomputer, der die Funktion der Relais erfüllt. Wie beim Datex-P-Netz stellt der Computer jetzt die Verbindung her. Im Gegensatz zur Relais-Vermittlung ist die Computervermittlung erheblich schneller und zuverlässiger, da die mechanische Vermittlung wegfällt.

## Gespräch im Binär-Code

Als nächstes werden die akustischen Signale, also die Sprache, in digitale Signale (wir kennen dies von den Spracheingabemodulen des C 64) umgesetzt. Diese digitalen Signale werden dann über eine virtuelle Verbindung ähnlich Datex-P übertragen.

Dieser Binärcode wird dann von der Vermittlungsstelle bei der Gegenstelle wieder in verständliche akustische Signale umgesetzt.

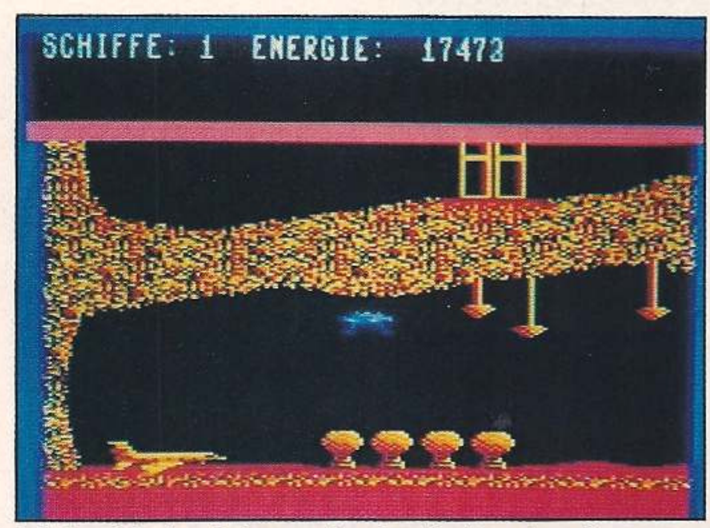
Wer nun meint, bei dieser Art der Vermittlung würden Verzögerungen auftreten, der täuscht sich. Die Datenübertragung erfolgt, wie bei Datex-P, mit bis zu 64000 Baud. Das reicht auch für den schnellsten Sprecher.

In Zukunft werden also Telefongespräche digital vermittelt und übertragen. Dies hat den Vorteil, daß wie bei Datex-P durch die virtuelle Verbindung wesentlich weniger Leitungen benötigt werden. Und dieselben Leitungen können auch für den Datenaustausch zwischen Computern verwendet werden. Telefongespräche, Datex-P, Bildschirmtext und die Telebox können in Zukunft direkt über dasselbe Kabel laufen.

Das ISDN-Netz soll ab 1988 zur Realität werden.

(rg)

Info: Die Informationsbroschüren der Bundespost sind über jedes Fernmeldeamt erhältlich.




# Die Scroll-Machine: Das Fenster zur Spielewelt




Bild 1 und 2. Ausschnitte aus dem Pseudoschirm eines Demos

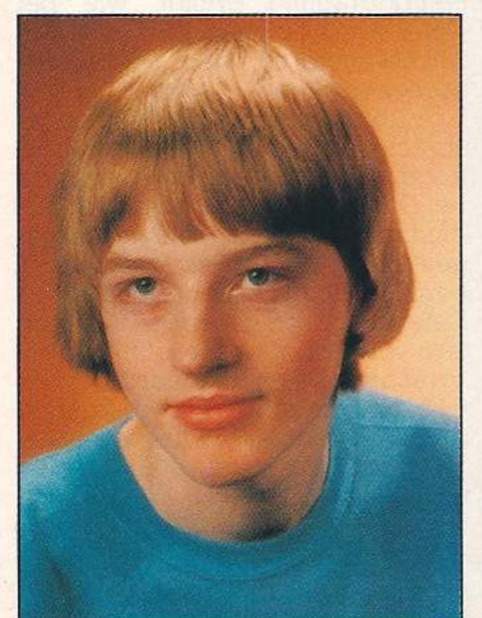
**Dieses außergewöhnliche Programm verschiebt den Bildschirm flimmerfrei in jede Richtung, wie es sonst nur von professionellen Spielen her bekannt ist — und das mit Basic.**

Bildschirmscrolling in eine beliebige Richtung ist im Commodore Basic V2.0 nicht realisierbar. Wer es dennoch versucht, wird mit dem Ergebnis nicht zufrieden sein. Dieses Programm macht es auch dem Anfänger leicht, Soft-Scrolling, ein Synonym für ruckfreies Verschieben des Bildschirms in alle Richtungen, in seine eigene Programme einzubauen. Besonders gut eignet es sich dazu mit dem Pseudoschirm-Editor Spielfelder zu konstruieren, die die Größe des normalen Bildschirms überschreiten. Durch eine große Anzahl von Befehlen ist jeder in der Lage, mittels Soft-Scroll immer den Teil des Feldes darzustellen, in dem sich die Spielfigur befindet. Die Spielfiguren werden ähnlich wie in professionellen Programmen durch Basic-Befehle, die die Beschleunigung, Reibung oder Höchstgeschwindigkeit festlegen, beliebig auf dem sichtbaren Pseudoschirm bewegt, einem Schirm, der in x- und y-Richtung größer ist als der normale Textbildschirm (x>40, y>25). Jeder kann maximal zwei Textfenster realisieren, die beliebige Spielinformationen enthalten. Das Programm stellt zum Laden und Abspeichern von SEQ- und PRG-Files einige sehr leistungsfähige Befehle zur Verfügung.

Aber nicht nur für die Gestaltung von Spielen ist dieses Programm geeignet. Der Anwender kann zum Beispiel von Basic aus Textverarbeitungsprogramme gestalten, die durch horizontales Scrollen mehr als 40 Zeichen pro Zeile darstellen können, wie es bei fast allen kommerziellen Programmen der Fall ist.

Bei dem Pseudoschirm-Editor handelt es sich um ein in Basic geschriebenes Programm, das durch einfache Befehle und einen geeigneten Zeichensatz beliebige Landschaften auf dem Bildschirm entstehen läßt. Die Konstruktion von Gängen und Höhlen wird im Kontur- und Micromodus zum Kinderspiel. Das Programm arbeitet sowohl mit einfarbigen wie aber auch mehrfarbigen Zeichensätzen. Die beiden Bilder sind ein ausgezeichnetes Beispiel für die Leistungsfähigkeit dieses Programms.

(Thilo Herrmann/ah)



## Lebenslauf

Ich wurde am 20.12.1968 in Göppingen geboren und ging nach erfolgreichem Besuch der Grundschule 1979 auf das Gymnasium, wo ich nun wenig begeistert die 10te Klasse besuche. Bis zum 14ten Lebensjahr stieg mein Interesse an Computern langsam an. Im Göppinger Computershop entdeckte ich einen TI99 4/A. Nach kurzer Zeit wurde mir das Herumexperimentieren mit diesem Computer zu wenig und ich entschloß mich, von meinen Ersparnissen einen VC 20 mit Datasette zu kaufen. Mit ihm war es mir möglich, das Commodore Basic zu erlernen. Ein halbes Jahr später stieß ich im Kaufhaus auf einen C 64 und war von seinen Fähigkeiten begeistert. So wurde der VC 20 kurzerhand zu Weihnachten gegen seinen größeren Bruder eingetauscht. Vom Rest meiner Ersparnisse schaffte ich mir ein 1541-Diskettenlaufwerk an. Nun konnte ich endlich richtig programmieren. Zuerst noch viel in Basic, doch dann immer mehr in Maschinensprache. Das Ergebnis meiner Bemühungen ist die Scroll-Machine, die mein bisher größtes Maschinenprogramm ist. Ohne Zweifel werden ihr noch größere Programme folgen. Es war nicht immer einfach, sämtliche Hürden zu überwinden, denn beim Programmieren des Videoprozessors sind viele Besonderheiten zu beachten, die nicht leicht zu verstehen und in keinem Buch zu finden sind. (Thilo Herrmann/ak)


64ER ONLINE

# MSE-Abtippen sicher und leicht gemacht

## Ähnlich wie der »Checksummer« ist auch der MSE ein Hilfsmittel bei der Eingabe von Listings, diesmal jedoch bei reinen Maschinensprache-Programmen.

Im Gegensatz zum »Checksummer« aber ist die Eingabe nicht ohne den MSE möglich. Der MSE verringert die Tipparbeit um ein Drittel und schließt Fehleingaben vollkommen aus. Außerdem können Sie die DATAs blind eingeben, ohne andauernd auf den Bildschirm schauen zu müssen. Dies wird durch akustische Meldungen realisiert.

MSE ist ein Maschinenspracheditor, mit dem ein Vertippen ausgeschlossen ist. Eine abgetippte Zeile wird nur angenommen, wenn sie richtig ist. Eine Checksumme am Ende jeder Zeile prüft, ob die richtigen Werte in der richtigen Zeile an der richtigen Stelle stehen. Wenn nicht, ertönt ein Warnsignal, und man beseitigt den Fehler.

War die Zeile korrekt, erklingt ein Gong, und die nächste Zeilennummer wird ausgegeben. Damit ist also auch »blindes« Eintippen möglich; Sie können sich voll auf den Text konzentrieren.

**So arbeitet man mit MSE**

Laden und starten Sie MSE. Zuerst wird der Programmname und die Start- und Endadresse erfragt. **Diese Angaben entnehmen Sie dem Kopf des jeweiligen abgedruckten Listings.** MSE meldet sich dann mit der Zeilennummer der ersten Zeile. Wenn Sie die Zeile richtig eingegeben haben, erscheint die nächste Zeilennummer und so weiter bis zum Ende. Zum Schluß wird das fertige Programm mit »CTRL-S« auf Diskette oder Kassette abgespeichert. Dazu sind keine weiteren Angaben mehr erforderlich. Das Programm kann dann ganz normal wieder geladen und gestartet werden. Wenn Sie nicht alles auf einmal tippen wollen, können Sie jederzeit unterbrechen und den eingetippten Teil mit »CTRL-S« abspeichern. Wollen Sie weiterarbeiten, laden und starten Sie MSE wieder.

Geben Sie auf die Frage nach der Startadresse aber jetzt »L« ein, um Ihr Teilprogramm zu laden. Jetzt können Sie mit »CTRL-N« die Adresse eingeben, an der Sie weitertippen müssen. Wenn Sie sich nicht gemerkt haben, wie weit Sie gekommen sind, geben Sie nach dem Laden »CTRL-M« ein.

Auf die Frage nach der Startadresse antworten Sie mit der Anfangsadresse, die links in der Kopfzeile auf dem Bildschirm steht. Nun wird Ihr Programm aufgelistet. Mit »SPACE« wird das Listen fortgesetzt, mit »STOP« abgebrochen. Das Ende Ihres Programmteils erkennen Sie sehr einfach daran, daß nur noch der Wert »AA« in der Zeile steht. Die Adresse dieser Zeile müssen Sie anschließend mit »CTRL-N« eingeben. Das Programm ist nur mit »STOP/RESTORE« zu verlassen. Speichern Sie aber vorher unbedingt immer Ihren Text ab.

**Hinweise zum Abtippen**

Vor dem Abtippen oder späteren Wiederladen des MSE-**Laders** müssen Sie unbedingt folgende Zeile eingeben:

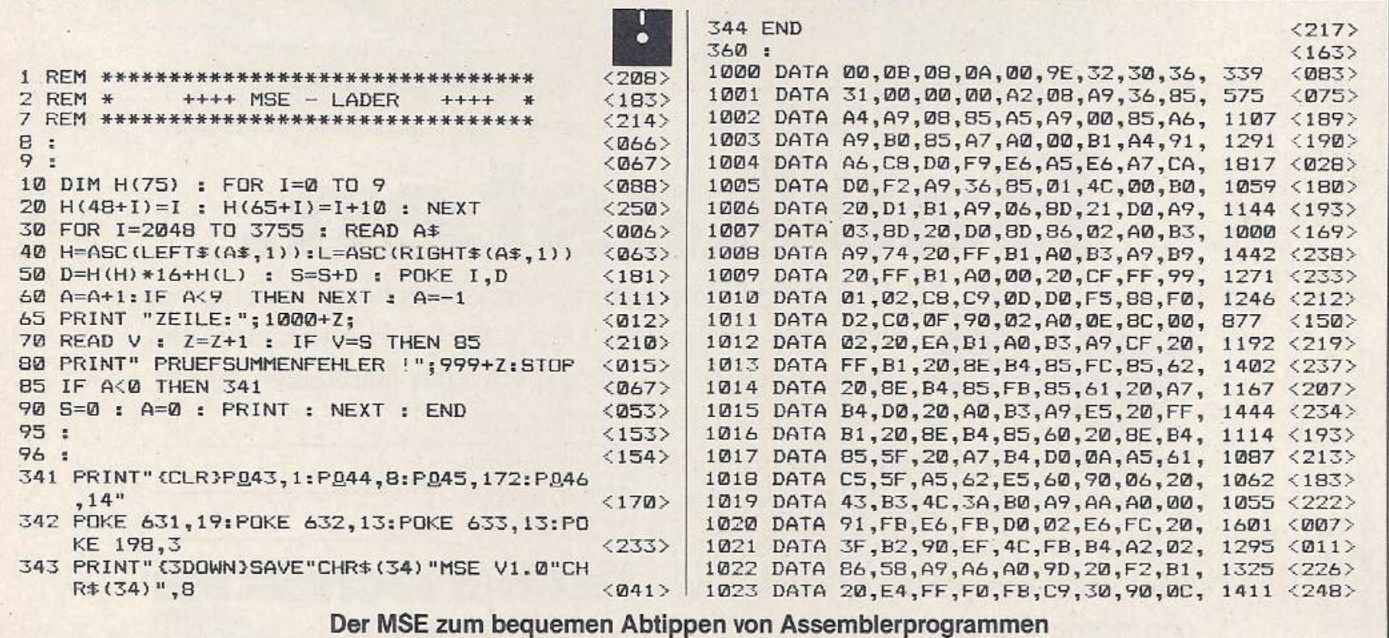
**POKE 43,1:POKE 44,32:POKE 8192,0 NEW**

Starten Sie das Programm mit RUN. Fehlerhafte Zeilen werden angezeigt und müssen korrigiert werden, bis der Lader zum »READY« durchläuft. Jetzt müssen Sie das fertige MSE-Programm abspeichern. Dazu brauchen Sie nur »RETURN« zu drücken, weil die erforderlichen Angaben schon auf dem Bildschirm stehen. (Kassettenbesitzer müssen in Zeile 343 die letzte Zahl in »1« abändern). Ab jetzt können Sie »MSE V1.0« direkt, also ohne den DATA-Lader, benutzen, MSE V1.0 wird ganz normal mit »,8« geladen (keine POKEs notwendig).

(N. Mann /D. Weineck/gk)

**MSE-Befehle:**

| | |
|---|---|
| DEL | löscht die letzt Eingabe. |
| CTRL-S | speichert das eingetippte Programm ab. |
| CTRL-L | lädt ein Programm. Start- und Endadresse werden automatisch ermittelt. |
| CTRL-M | listet den Speicherinhalt. Abbruch mit STOP-Taste, weiter mit Leertaste. |
| CTRL-N | erlaubt die Eingabe einer neuen Adresse zum Weitertippen. |
| CTRL-P | gibt ein MSE-Listing auf dem Drucker aus. |



```
1 REM ********************************      <208>
2 REM *      ++++ MSE - LADER   ++++  *      <183>
7 REM ********************************      <214>
8 :                                          <066>
9 :                                          <067>
10 DIM H(75) : FOR I=0 TO 9                  <088>
20 H(48+I)=I : H(65+I)=I+10 : NEXT           <250>
30 FOR I=2048 TO 3755 : READ A$              <006>
40 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))    <063>
50 D=H(H)*16+H(L) : S=S+D : POKE I,D         <181>
60 A=A+1:IF A<9  THEN NEXT : A=-1            <111>
65 PRINT "ZEILE:";1000+Z;                    <012>
70 READ V : Z=Z+1 : IF V=S THEN 85           <210>
80 PRINT" PRUEFSUMMENFEHLER !";999+Z:STOP    <015>
85 IF A<0 THEN 341                           <067>
90 S=0 : A=0 : PRINT : NEXT : END            <053>
95 :                                         <153>
96 :                                         <154>
341 PRINT"{CLR}P043,1:P044,8:P045,172:P046
    ,14"                                     <170>
342 POKE 631,19:POKE 632,13:POKE 633,13:PO
    KE 198,3                                 <233>
343 PRINT"{3DOWN}SAVE"CHR$(34)"MSE V1.0"CH
    R$(34)",8                                <041>
344 END                                      <217>
360 :                                        <163>
1000 DATA 00,0B,08,0A,00,9E,32,30,36, 339  <083>
1001 DATA 31,00,00,00,A2,08,A9,36,85, 575  <075>
1002 DATA A4,A9,08,85,A5,A9,00,85,A6, 1107 <189>
1003 DATA A9,B0,85,A7,A0,00,B1,A4,91, 1291 <190>
1004 DATA A6,C8,D0,F9,E6,A5,E6,A7,CA, 1817 <028>
1005 DATA D0,F2,A9,36,85,01,4C,00,B0, 1059 <180>
1006 DATA 20,D1,B1,A9,06,8D,21,D0,A9, 1144 <193>
1007 DATA 03,8D,20,D0,8D,86,02,A0,B3, 1000 <169>
1008 DATA A9,74,20,FF,B1,A0,B3,A9,B9, 1442 <238>
1009 DATA 20,FF,B1,A0,00,20,CF,FF,99, 1271 <233>
1010 DATA 01,02,C8,C9,0D,D0,F5,88,F0, 1246 <212>
1011 DATA D2,C0,0F,90,02,A0,0E,8C,00, 877  <150>
1012 DATA 02,20,EA,B1,A0,B3,A9,CF,20, 1192 <219>
1013 DATA FF,B1,20,8E,B4,85,FC,85,62, 1402 <237>
1014 DATA 20,8E,B4,85,FB,85,61,20,A7, 1167 <207>
1015 DATA B4,D0,20,A0,B3,A9,E5,20,FF, 1444 <234>
1016 DATA B1,20,8E,B4,85,60,20,8E,B4, 1114 <193>
1017 DATA 85,5F,20,A7,B4,D0,0A,A5,61, 1087 <213>
1018 DATA C5,5F,A5,62,E5,60,90,06,20, 1062 <183>
1019 DATA 43,B3,4C,3A,B0,A9,AA,A0,00, 1055 <222>
1020 DATA 91,FB,E6,FB,D0,02,E6,FC,20, 1601 <007>
1021 DATA 3F,B2,90,EF,4C,FB,B4,A2,02, 1295 <011>
1022 DATA 86,58,A9,A6,A0,9D,20,F2,B1, 1325 <226>
1023 DATA 20,E4,FF,F0,FB,C9,30,90,0C, 1411 <248>
```

**Der MSE zum bequemen Abtippen von Assemblerprogrammen**

```
1024 DATA C9,47,B0,08,C9,3A,90,0B,C9, 1071 <228>
1025 DATA 41,B0,07,C9,14,D0,0F,4C,0B, 779  <173>
1026 DATA B1,20,D2,FF,A6,58,95,F7,C6, 1522 <254>
1027 DATA 58,D0,D2,60,AE,8D,02,F0,26, 1197 <231>
1028 DATA C9,0C,D0,03,4C,0B,B6,C9,13, 913  <187>
1029 DATA D0,03,4C,8B,B5,C9,0D,D0,03, 1032 <228>
1030 DATA 4C,BA,B4,C9,10,D0,03,4C,68, 1050 <232>
1031 DATA B5,C9,0E,D0,06,20,5F,B4,4C, 993  <203>
1032 DATA 64,B1,4C,92,B0,A5,F9,20,02, 1123 <204>
1033 DATA B1,0A,0A,0A,0A,85,F9,A5,F8, 1012 <247>
1034 DATA 20,02,B1,05,F9,60,C9,3A,90, 964  <154>
1035 DATA 02,69,08,29,0F,60,A6,59,E0, 746  <153>
1036 DATA 08,90,1F,A6,58,E0,02,B0,06, 845  <155>
1037 DATA 20,D2,FF,4C,8E,B0,C6,59,A0, 1338 <017>
1038 DATA 14,A9,92,20,F2,B1,CA,D0,FA, 1446 <006>
1039 DATA 84,57,68,68,4C,8B,B1,A6,D3, 1196 <249>
1040 DATA E0,08,B0,03,4C,92,B0,20,D2, 1051 <200>
1041 DATA FF,A6,58,E0,02,90,09,C6,59, 1175 <242>
1042 DATA 20,D2,FF,C6,58,D0,F9,4C,8E, 1458 <040>
1043 DATA B0,48,4A,4A,4A,4A,20,59,B1, 842  <185>
1044 DATA 68,29,0F,C9,0A,90,02,69,06, 628  <163>
1045 DATA 69,30,4C,D2,FF,A2,FC,9A,20, 1294 <027>
1046 DATA D1,B1,20,48,B2,20,EA,B1,20, 1143 <217>
1047 DATA 9F,B2,A5,FC,20,4E,B1,A5,FB, 1457 <057>
1048 DATA 20,4E,B1,20,ED,B1,A9,3A,A0, 1120 <250>
1049 DATA 20,20,F2,B1,A9,00,85,59,20, 906  <145>
1050 DATA 8E,B0,20,ED,B1,A4,59,20,EF, 1288 <029>
1051 DATA B0,91,FB,C8,84,59,C0,08,90, 1337 <249>
1052 DATA EC,20,10,B2,A9,12,20,D2,FF, 1146 <251>
1053 DATA 20,8E,B0,20,EF,B0,C5,FF,F0, 1489 <048>
1054 DATA 0D,20,43,B3,A9,14,A0,14,20, 692  <155>
1055 DATA F2,B1,4C,A2,B1,A9,92,20,D2, 1391 <005>
1056 DATA FF,20,33,B2,20,E0,B2,20,3F, 1045 <203>
1057 DATA B2,90,9F,4C,8B,B5,A9,93,20, 1225 <010>
1058 DATA D2,FF,A2,00,A9,03,9D,00,D8, 1172 <011>
1059 DATA 9D,00,D9,9D,00,DA,9D,00,DB, 1125 <030>
1060 DATA E8,D0,EF,60,A9,0D,2C,A9,20, 1202 <029>
1061 DATA 4C,D2,FF,20,D2,FF,98,4C,D2, 1476 <069>
1062 DATA FF,20,E4,FF,F0,FB,60,84,5D, 1582 <069>
1063 DATA 85,5C,A0,00,B1,5C,F0,06,20, 932  <183>
1064 DATA D2,FF,C8,D0,F6,60,A5,FB,85, 1764 <075>
1065 DATA 5A,A0,00,84,5B,B1,FB,18,65, 1026 <254>
1066 DATA 5A,85,5A,90,02,E6,5B,06,5A, 876  <212>
1067 DATA 26,5B,C8,C0,08,90,EC,A5,5A, 1164 <028>
1068 DATA 65,5B,85,FF,60,18,A5,FB,69, 1221 <028>
1069 DATA 08,85,FB,90,02,E6,FC,60,A5, 1281 <020>
1070 DATA FB,C5,5F,A5,FC,E5,60,60,A0, 1541 <061>
1071 DATA B3,A9,FB,20,FF,B1,A0,01,B9, 1409 <053>
1072 DATA 00,02,20,D2,FF,CC,00,02,C8, 905  <202>
1073 DATA 90,F4,A9,10,ED,00,02,AA,20, 1014 <247>
1074 DATA ED,B1,CA,D0,FA,A5,62,20,4E, 1447 <073>
1075 DATA B1,A5,61,20,4E,B1,20,ED,B1, 1172 <013>
1076 DATA A5,60,20,4E,B1,A5,5F,20,4E, 918  <223>
1077 DATA B1,A9,9F,20,D2,FF,20,EA,B1, 1445 <065>
1078 DATA 24,5E,10,01,60,A9,12,20,D2, 672  <165>
1079 DATA FF,A2,28,20,ED,B1,CA,D0,FA, 1563 <095>
1080 DATA A9,92,4C,D2,FF,A5,D6,C9,16, 1458 <078>
1081 DATA B0,01,60,A9,A0,85,A4,A9,78, 1188 <013>
1082 DATA 85,A6,A9,04,85,A5,85,A7,A2, 1232 <018>
1083 DATA 13,A0,27,B1,A4,91,A6,88,10, 1022 <235>
1084 DATA F9,CA,F0,19,18,A5,A4,69,28, 1214 <039>
1085 DATA 85,A4,90,02,E6,A5,18,A5,A6, 1193 <018>
1086 DATA 69,28,85,A6,90,E0,E6,A7,4C, 1285 <038>
1087 DATA B6,B2,A9,91,4C,D2,FF,A9,0F, 1399 <097>
1088 DATA 8D,18,D4,A9,00,8D,05,D4,A9, 1073 <040>
1089 DATA F7,8D,06,D4,A9,11,8D,04,D4, 1149 <046>
1090 DATA A9,32,8D,01,D4,A9,00,8D,00, 883  <226>
1091 DATA D4,A0,80,20,09,B3,A9,10,8D, 1046 <009>
1092 DATA 04,D4,60,A2,FF,CA,D0,FD,88, 1528 <090>
1093 DATA D0,F8,60,A9,0F,8D,18,D4,A9, 1282 <068>
1094 DATA 2D,8D,05,D4,A9,A5,8D,06,D4, 1096 <066>
1095 DATA A9,21,8D,04,D4,A9,07,8D,01, 877  <243>
1096 DATA D4,A9,05,8D,00,D4,A0,FF,20, 1186 <053>
1097 DATA 09,B3,A9,20,8D,04,D4,A9,00, 915  <231>
1098 DATA 8D,01,D4,8D,00,D4,60,38,20, 891  <219>
1099 DATA F0,FF,8A,48,98,48,18,A0,06, 1119 <046>
1100 DATA A2,18,20,F0,FF,A0,B4,A9,0A, 1232 <057>
1101 DATA 20,FF,B1,20,12,B3,20,E4,FF, 1208 <045>
1102 DATA F0,FB,A2,1D,A9,14,20,D2,FF, 1368 <092>
1103 DATA CA,D0,FA,68,A8,68,AA,18,4C, 1306 <098>
1104 DATA F0,FF,0D,0D,0D,20,20,20,20, 662  <231>
1105 DATA 20,20,20,4D,41,53,43,48,49, 533  <170>
1106 DATA 4E,45,4E,53,50,52,41,43,48, 674  <205>
1107 DATA 45,20,2D,20,45,44,49,54,4F, 551  <197>
1108 DATA 52,20,0D,0D,20,20,20,20,20, 300  <149>
1109 DATA 20,20,20,56,4F,4E,20,4E,2E, 495  <224>
1110 DATA 4D,41,4E,4E,20,26,20,44,2E, 514  <221>
1111 DATA 57,45,49,4E,45,43,4B,00,0D, 531  <216>
1112 DATA 0D,0D,20,20,20,50,52,4F,47, 434  <197>
1113 DATA 52,41,4D,4D,4E,41,4D,45,20, 622  <227>
1114 DATA 3A,20,00,0D,0D,20,20,20,53, 295  <185>
1115 DATA 54,41,52,54,41,44,52,45,53, 682  <177>
1116 DATA 53,45,20,3A,20,24,00,0D,0D, 336  <194>
1117 DATA 20,20,20,45,4E,44,41,44,52, 526  <177>
1118 DATA 45,53,53,45,20,20,20,3A,20, 490  <172>
1119 DATA 24,00,92,05,20,50,52,4F,47, 531  <180>
1120 DATA 52,41,4D,4D,20,3A,20,00,12, 441  <195>
1121 DATA 20,20,2A,2A,2A,20,46,41,4C, 433  <211>
1122 DATA 53,43,48,45,20,45,49,4E,47, 614  <208>
1123 DATA 41,42,45,20,2A,2A,2A,20,20, 422  <193>
1124 DATA 92,00,0D,0D,2A,2A,2A,20,45, 399  <243>
1125 DATA 4E,44,45,20,2A,2A,2A,00,13, 392  <223>
1126 DATA 05,20,20,12,44,92,49,53,4B, 532  <189>
1127 DATA 20,4F,44,45,52,20,12,54,92, 610  <190>
1128 DATA 41,50,45,0D,00,13,20,20,49, 383  <181>
1129 DATA 2F,4F,20,2D,20,46,45,48,4C, 522  <247>
1130 DATA 45,52,00,20,D1,B1,20,48,B2, 851  <215>
1131 DATA A0,B3,A9,CF,20,FF,B1,20,8E, 1353 <115>
1132 DATA B4,85,FC,20,8E,B4,85,FB,C5, 1500 <115>
1133 DATA 61,A5,FC,E5,62,90,23,A5,FB, 1436 <098>
1134 DATA C5,5F,A5,FC,E5,60,B0,19,20, 1267 <097>
1135 DATA A7,B4,D0,14,60,20,A7,B4,F0, 1290 <065>
1136 DATA 0C,85,F9,20,A7,B4,F0,05,85, 1151 <066>
1137 DATA F8,4C,EF,B0,68,68,20,43,B3, 1225 <090>
1138 DATA 4C,5F,B4,20,CF,FF,C9,4C,D0, 1330 <146>
1139 DATA 09,20,D1,B1,20,48,B2,4C,0B, 796  <010>
1140 DATA B6,C9,0D,60,A9,00,85,5E,20, 920  <019>
1141 DATA 5F,B4,20,EA,B1,20,0D,B5,24, 980  <040>
1142 DATA 5E,30,05,20,E4,FF,F0,FB,20, 1185 <097>
1143 DATA E1,FF,F0,26,20,9F,B2,24,5E, 1257 <110>
1144 DATA 10,09,20,4E,B5,20,0D,B5,20, 574  <246>
1145 DATA 60,B5,20,33,B2,20,3F,B2,90, 955  <000>
1146 DATA D7,A0,B4,A9,28,20,FF,B1,20, 1260 <095>
1147 DATA E4,FF,C9,0D,D0,F9,A9,00,85, 1456 <141>
1148 DATA 5E,A5,61,85,FB,A5,62,85,FC, 1388 <131>
1149 DATA 20,E0,B2,4C,64,B1,A5,FC,20, 1236 <092>
1150 DATA 4E,B1,A5,FB,85,FF,20,4E,B1, 1346 <144>
1151 DATA A9,20,A0,3A,20,F2,B1,A0,00, 1030 <053>
1152 DATA 20,ED,B1,B1,FB,20,4E,B1,C8, 1361 <128>
1153 DATA C0,08,90,F3,20,ED,B1,24,5E, 1163 <090>
1154 DATA 30,03,A9,12,2C,A9,20,20,D2, 725  <255>
1155 DATA FF,20,10,B2,A5,FF,20,4E,B1, 1188 <123>
1156 DATA A9,92,20,D2,FF,4C,EA,B1,A9, 1468 <157>
1157 DATA FF,85,B8,85,B9,A9,04,85,BA, 1382 <143>
1158 DATA 20,C0,FF,A2,FF,4C,C9,FF,20, 1460 <165>
1159 DATA CC,FF,A9,FF,4C,C3,FF,20,5F, 1536 <215>
1160 DATA B4,A9,80,85,5E,20,4E,B5,20, 1027 <088>
1161 DATA 48,B2,A2,24,A9,2D,20,D2,FF, 1159 <121>
1162 DATA CA,D0,FA,20,EA,B1,20,EA,B1, 1546 <162>
1163 DATA 20,60,B5,4C,C1,B4,20,B8,B5, 1155 <093>
1164 DATA A6,5F,A4,60,A9,61,20,D8,FF, 1290 <131>
1165 DATA B0,0A,20,B7,FF,29,BF,D0,03, 1099 <135>
1166 DATA 4C,FB,B4,A9,01,20,C3,FF,20, 1191 <131>
1167 DATA 68,B6,A0,B4,A9,4F,20,FF,B1, 1338 <147>
1168 DATA 20,F9,B1,4C,FB,B4,20,68,B6, 1283 <130>
1169 DATA A9,37,A0,B4,20,FF,B1,20,F9, 1309 <126>
1170 DATA B1,A2,08,C9,44,F0,06,A2,01, 1025 <079>
1171 DATA C9,54,D0,F1,A9,01,A8,20,BA, 1290 <123>
1172 DATA FF,A0,00,E0,01,F0,1A,A9,40, 1139 <110>
1173 DATA 8D,20,02,A9,3A,8D,21,02,B9, 763  <050>
1174 DATA 01,02,99,22,02,C8,CC,00,02, 598  <013>
1175 DATA 90,F4,C8,C8,D0,0C,B9,01,02, 1196 <121>
1176 DATA 99,20,02,C8,CC,00,02,D0,F4, 1045 <092>
1177 DATA 98,A2,20,A0,02,4C,BD,FF,20, 1060 <119>
1178 DATA B8,B5,A5,BA,C9,08,90,33,A6, 1286 <146>
1179 DATA B9,86,57,A9,01,20,C3,FF,A9, 1227 <136>
1180 DATA 60,85,B9,20,C0,FF,B0,28,A5, 1274 <126>
1181 DATA BA,20,B4,FF,A5,B9,20,96,FF, 1440 <174>
1182 DATA 20,A5,FF,85,61,A5,90,4A,4A, 1139 <128>
1183 DATA B0,13,20,A5,FF,85,62,20,AB, 1081 <112>
1184 DATA FF,A5,57,85,B9,A9,00,20,D5, 1239 <141>
1185 DATA FF,90,03,4C,A3,B5,86,5F,84, 1183 <144>
1186 DATA 60,A5,BA,C9,01,D0,0A,AD,3D, 1101 <149>
1187 DATA 03,85,61,AD,3E,03,85,62,4C, 778  <063>
1188 DATA FB,B4,A9,13,20,D2,FF,A2,1C, 1306 <168>
1189 DATA 20,ED,B1,CA,D0,FA,60, 1202      <104>
```

**Listing von MSE (Schluß)**

# Scroll-Machine: Das Fenster zur Spielewelt

**Bringen Sie Bewegung in Ihre Grafik. Nutzen Sie einen Bildschirm, der bis zu 30 mal größer ist als der normale Textbildschirm. Scrolling wie bei dem Spiel Zeppelin wird dadurch möglich.**

Das Programm besteht aus 3 KByte Maschinensprache, die im Bereich von $C000 (49152) bis fast $CBFF (52223) liegen. Die Variablen stehen von 828 bis 959 im Kassettenpuffer. Das Video-RAM, das vom Programm benutzt wird, liegt im Bereich von 52224 bis 53223. Da die Spritepointer immer dem Video-RAM folgen, stehen sie nun von 53240 bis 53247 (normal: 2040 bis 2047). Im Bereich von $D000 (53248) bis $D7FF (55295) steht der Zeichensatz, der leicht verändert werden kann. Dabei ist zu beachten, daß der Bereich von $D000 bis $DFFF dreimal belegt ist, einmal durch die Ein-/Ausgabe (VIC, SID, CIA 1 und 2), den C 64-Zeichensatz und das RAM, in das der Zeichensatz kopiert oder ein neuer geladen wird.

Außerdem benötigt die Scroll-Machine noch eine Multiplikationstabelle, die von $DE00 (56832) bis $DFFF (57343) steht. Der Rest von $D800 bis $DDFF (55296 bis 56831) kann von Sprites genutzt werden. Ebenfalls für Sprites frei ist das RAM unter dem Kernal von $E000 bis $FFFF (57344 bis 65535). Dort kann auch ein 8-KByte-Pseudoschirm stehen, der eine größere X/Y-Ausdehnung hat als der normale Textbildschirm. Wenn dieser Bereich jedoch frei ist, kann man insgesamt 152 verschiedene Sprites im Speicher halten. Es muß aufgrund einer Besonderheit des VICs darauf geachtet werden, daß die Speicherstelle 65535 den Wert Null enthält. Die Sprites müssen nach wie vor vom Basic-Programm bewegt werden.

Es sei nicht unerwähnt, daß ich die Restore-Taste geändert habe. Sie funktioniert nun auch ohne daß man RUN-STOP drückt. Die NMI-Vektoren $0318/$0319 (792/793) und $FFFA/$FFFB (65530/65531) im RAM enthalten immer die Adresse der neuen Restore-Routine. Soll die Restore-Taste gesperrt werden, so hilft POKE 51692,64. Die RUN/STOP-Taste kann mit POKE 49710,52 außer Kraft gesetzt werden (TI wird damit auch unbrauchbar).

## Die Funktionsweise

Das Programm arbeitet mit drei Rasterzeileninterrupts, zwei für den oberen Text/Soft-Scroll-Übergang und einer für den unteren. Alle Interrupts können mit dem RZ-Befehl verschoben werden, so daß die Bildschirmfenster beliebig groß werden können.

Jede der drei Interruptroutinen setzt erst die Parameter des Videoprozessors: Hintergrundfarbe, Rahmenfarbe und so weiter. Die erste Interruptroutine hat damit ihre Aufgabe erfüllt. Sie setzt die Hintergrundfarbe auf die Zeichenfarbe, damit ein sauberer Übergang zwischen Soft-Scroll und normalem Text entsteht.

Die zweite Routine berechnet die Werte für Bewegung und Steuerung. Dann springt sie zum normalen Systeminterrupt, der die Tastatur abfragt und die Zeit erhöht.

Der dritte Interrupt ruft die Scroll-Routine auf, die den Pseudobildschirm abtastet und am meisten Zeit verschlingt.

Dieser Teil des Programms muß sehr schnell sein, da er seine Aufgabe erledigt haben muß, bevor der nächste Interrupt ausgelöst wird.

Die neuen Basic-Befehle werden mit Hilfe des Pointers $0308/$0309 eingebunden.

TI$ wird unbrauchbar, da der Systeminterrupt nun nicht mehr jede 60stel Sekunde, sondern jede 50stel Sekunde ausgeführt wird. Die Sekunde errechnet sich während des Betriebs des Soft-Scrolls nun mit TI/50.

## Joysticksteuerung

Wie sich aus der Befehlsliste ersehen läßt, besitzt das Programm vier Befehle zur Steuerung des Joysticks.

Es gibt mehrere Möglichkeiten, ein Objekt zu steuern. Zum einen gibt es die »lineare« Steuerung. Sie wird wohl am häufigsten verwendet, zum Beispiel bei Fort Apocalypse und den meisten selbstgeschriebenen Spielen. Das funktioniert so: Man drückt den Steuerknüppel in eine bestimmte Richtung und das Objekt setzt sich in Bewegung. Es bewegt sich so lange, bis man den Knüppel wieder losläßt. Diese Steuerung ist jedoch wenig wirklichkeitsnah, man denke nur an eine Raumkapsel, die durch Raketenschübe gesteuert wird. Sie setzt sich nicht sprunghaft in Bewegung, sondern langsam. Wenn der Antrieb dann abgeschaltet wird, stoppt unser Flugkörper nicht plötzlich, sondern fliegt mit der gleichen Geschwindigkeit weiter. Diese träge Steuerung und die lineare Steuerung lassen sich mit der Scroll-Machine verwirklichen.

Da die lineare Steuerung besonders einfach ist, werden ihre Werte im JS-Befehl mit angegeben. Nämlich so:

**← JS m,x,y,p** (m = Joystickmodus, p = Port)

x und y stellen die Geschwindigkeiten der linearen Steuerung dar. Bei x = 8 und y = 0 hat man in etwa die Space-Invader-Einstellung.

Die träge Steuerung benötigt drei Befehle, was schon andeutet, daß sie etwas kompliziert ist. Hier bestimmt man nicht die Geschwindigkeit, sondern die Beschleunigung. Da vor jedem Trägheitsbefehl ein T steht, lautet der Befehl:

**←TB x,y**

x und y geben die X-Y-Beschleunigung an. Sie können Werte zwischen Null und 255 annehmen.

Wenn man nun den Steuerknüppel in eine Richtung drückt, wird die Geschwindigkeit immer größer. Die größtmögliche Geschwindigkeit kann man mit dem TG-Befehl festlegen.

**←TG x, y**

Dies kann man sehr genau regeln, weil x und y Werte zwischen Null und 32767 annehmen können.

Wenn man einen Gegenstand auf einer waagrechten Fläche beschleunigt, so hält dieser Gegenstand irgendwann auch wieder an. Daran ist die Reibung schuld. Das kann man auch simulieren, der Befehl lautet:

**←TR x,y**

x und y müssen kleiner sein als die entsprechenden Werte beim TB-Befehl. Die Routine zieht x und y einfach von der Geschwindigkeit ab.

Natürlich kann man die lineare und die träge Steuerung auch mischen. (Es muß vor jedem Befehl ein Pfeil stehen.)
Zum Beispiel: Defender: ←RS: ←CP: ←ZF 1: ←US 1.
Diese Befehle setzen die Standardeinstellung.

| Befehl | Wert | Bedeutung |
|---|---|---|
| ← JS 2,0,8,2 | m=2 | — mischen |
| | x=0 | — keine lineare X-Steuerung |
| | y=8 | — lineare Y-Steuerung, Geschwindigkeit 8 |
| | p=2 | — Port 2 |
| ←TB 16,0 | x=16 | — X-Beschleunigung 16 |
| | y=0 | — keine Y-Beschleunigung |
| ←TG 2000,0 | x=2000 | — X-Geschwindigkeit höchstens 2000 |
| | y=0 | — Y-Geschwindigkeit 0 (hier unwichtig) |
| ←TR 6,0 | x=6 | — X-Reibung 6 |
| | y=0 | — Y-Reibung 0 (auch unwichtig) |

In diese Steuerung kann man auch noch Bewegung einmischen mit:

| Befehl | Wert | Bedeutung |
|---|---|---|
| ← BW 0,—8 | x=0 | — keine X-Bewegung |
| | y=—8 | — Mit Geschwindigkeit 8 nach oben. |

Ergebnis: Der Bildschirm bewegt sich kontinuierlich nach oben. Mit dem Joystick kann man die Y-Geschwindigkeit erhöhen oder vermindern.

## Die Befehle

Vor jedem Befehl steht der Pfeil nach links. Die folgenden zwei Buchstaben stellen das Befehlswort dar. Danach muß man teilweise einige Parameter angeben, wie zum Beispiel beim normalen POKE-Befehl. Nach einer IF...THEN-Entscheidung muß nach dem THEN ein Doppelpunkt stehen, wenn ein Scroll-Machine-Befehl folgt, zum Beispiel IF A=1 THEN: ← RS. Wem der Pfeil nicht paßt, der kann mit POKE 50550,ASC(X$) ein anderes Zeichen auswählen. So wird nach POKE 50550,ASC("!") nur noch das Ausrufezeichen angenommen. Folgende Zeichen lassen sich verwenden: !"#$%&'(),.:; sowie Klammeraffe, die eckigen Klammern, Pfund, PI, Pfeil links sowie fast alle Buchstaben.

Damit man das Programm auch von Maschinensprache aus verwenden kann, habe ich jeweils die entsprechenden Adressen und gegebenenfalls auch die Einsprungadressen mit angegeben.

Die Befehle:

**RS**

Reset. Dieser Befehl sollte am Anfang jedes Programms stehen. Er löscht alle Einstellungen und schaltet auf den Normalzustand, das heißt, er führt unter anderem folgende Befehle aus:
MT 40960,40,0: VI 52224,21
ML: JSR $C5C0

**MT aa,xd,yd**

Erstellt die Multiplikationstabelle, das heißt, man legt mit diesem Befehl die Anfangsadresse und die X- und Y-Ausdehnung des Pseudoschirms fest.
aa = Anfangsadresse, xd = Ausdehnung des Schirms in X-Richtung, yd = Ausdehnung in Y-Richtung.

Der normale Videoschirm hätte zum Beispiel die X- und Y-Werte 40 und 25. Die des Pseudoschirms können theoretisch zwischen Null und 8191 liegen.

Die Y-Ausdehnung ist völlig unwichtig. Sie gehört jedoch der Vollständigkeit halber dazu, wird mitgespeichert und -geladen und kann über USR(12) abgefragt werden.

Wenn der Schirm im Basic-RAM liegt ($aa < 40960$), sollte man das Basic-Ende so herabsetzen:
AA = Startadresse aa: H = INT(AA/256): L=AA-H*256: POKE 55,L: POKE 56,H: CLR
ML: aa:$CBFA/CBFB, xd:$CBFC/CBFD, yd:$CBFE/CBFF, JSR $C6AC

**CP**

Kopiert den Zeichensatz vom ROM in das darunterliegende RAM. Er führt UT 53248, 55295, 53248,51,48 aus. Dieser Befehl sollte ebenfalls am Anfang eines Programms stehen, falls nicht ein neu definierter Zeichensatz mit dem LD-Befehl geladen wurde.
ML: JSR $C8BC

**US n**

Umschalten zwischen Soft-Scroll (n=1) und Text (n=0). Bei n=0 werden alle Einstellungen des Textfensters gesetzt.
ML: LDX #n: JSR $C5F8

**ZF f**

Dient zum Setzen der Zeichenfarbe f. Wenn man im Multicolormodus arbeitet, kann man nur die Farben Null bis Sieben nutzen und muß jeweils Acht dazuzählen (Bit 3 setzen).
ML : LDA #n: JSR $C8EB

**HF f1,f2,f3**

Wählt die Hintergrundfarben der drei Bereiche.
f1= Hintergrundfarbe des Textfensters,
f2 = Hintergrundfarbe des Streifens. f2 muß gleich der Zeichenfarbe sein, damit ein sauberer Übergang entsteht.
f3 = Hintergrundfarbe des Soft-Scroll-Bereiches.
ML: $C568/$C569/$C56A

**RF f1,f2,f3**

Die Randfarben können beliebig gewählt werden.
ML: $C56B/$C56C/$C56D

**PO x,y**

Positioniert den Bildschirm innerhalb des Pseudoschirms. x und y können negativ und positiv sein.
Sinnvoll sind jedoch nur $0 < x < xd*8$ und $0 < y < yd*8$.
ML: x:$0358/$0359, y:$035A/$035B

**BW x,y**

Bewegt den Bildschirm in der angegebenen X- und Y-Geschwindigkeit. x und y können zwischen -127 und 128 liegen.
ML: x:$0385, y:$0386

**JS m,x,y,p**

Bestimmt die Werte für den Joystick.
m = Modus.
Bei m=0 keine Steuerung.
Bei m=1 lineare Steuerung.
Bei m=2 lineare und träge Steuerung.
x und y geben die Geschwindigkeit der linearen Steuerung an. Sie kann zwischen 0 und 255 liegen.
p steht für den Joystickport.
Die letzten drei Angaben können weggelassen werden.
ML: n:$033E, x:$0374, y:$0375, p:$03A5

Zur trägen Steuerung gibt es drei Befehle. Sie bewirken nur etwas, wenn der Joystickmodus zwei gewählt wurde. Der erste Buchstabe lautet immer »T«.

**TB x,y**

Regelt die Beschleunigung. x und y kann zwischen 0 und 255 liegen.
ML: x:$034A, y:$034B

**TR x,y**

Bestimmt die Reibung. x und y sind auch hier 1-Byte-Werte.
ML: x:$0392, y:$0393

**TG x,y**

Legt die Höchstgeschwindigkeit fest. x und y sind hier 2-Byte-Werte, sie können also Werte zwischen 0 und 65535 annehmen.
ML: x:$038E/$038F, y:$0390/$0391

Die nachfolgenden Befehle sind allgemeingültig:

**GR x,x2,y,y2**

Die hier festgelegten Grenzen können nicht überschritten

werden. Die Werte können wie beim PO-Befehl zwischen 0 und 65535 liegen.
ML: x:$0396/$0397, x2:$0398/$0399, y:$039A/$039B, y2:$039C/$039D

**GM m**

Der Grenzmodus wird gewählt: Bei m=0 bestehen keine Grenzen. Bei m=1 wird bei den Grenzen abgestoppt. Bei m=2 wird der Bildschirm an die gegenüberliegende Grenze gesetzt.
ML: m:$039E

**AS n**

Läßt den Interrupt n mal aussetzen, bevor er wieder ausgeführt wird. Das ist besonders nützlich, wenn man Rechenzeit sparen will.
Nachteil: Je größer n ist, desto stärker ruckt das Bild.
ML: LDX #n: JSR $C78D

**TM m**

Wählt den »Tastmodus«. Bei m=0 wird der Pseudoschirm nur abgetastet, wenn es nötig ist.
Bei m=1 wird er immer abgetastet (= jede 50stel Sekunde), was natürlich Rechenzeit kostet.
Nützlich, wenn im Pseudoschirm laufend etwas verändert wird. Ein Beispiel: RS:CP:ZF 1:MT 1024,40,25:PO 0,0:TM 1:US 1
PRINT "<HO>"
Ergebnis: Der blinkende Cursor erscheint doppelt auf dem Schirm. Der zweite Cursor macht alle Bewegungen mit. Wenn man TM 0 eingibt, ändert sich das.
ML: m:$03A2

**SY n**

Synchronisiert die Scroll-Machine mit dem laufenden Programm, das heißt, der Interrupt wird nicht mehr im 50stel-Sekunden-Takt ausgeführt, sondern im Programmtakt.
n=1 schaltet Sync-Modus ein und führt den nächsten Interrupt aus.
n=0 schaltet den Sync-Modus aus.
Diese Funktion ist wichtig bei langsamen Basic-Programmen wie dem Editor. Achtung: Die X/Y-Position wird erst beim darauffolgenden SY 1-Befehl gesetzt.
ML: LDX #n: JSR $C88C

**VI aa,z**

Verlegt das Video-RAM, in das geschrieben wird.
aa = neue Anfangsadresse, z = Anzahl der Zeilen (z <= 21).
Normal: aa= 52224, z=21. Je kleiner z ist, desto schneller wird das Programm.
ML: aa:$C4D9/$C4DA, LDX #z: JSR $C7AE

**1R n1,n2,n3**

Inhalt des 1. Steuerregisters (VIC+17) in den drei Bereichen. Normal: 27,23,16
ML: $C55C/$C55D/$C55E

**2R n1,n2,n3**

Der entsprechende Befehl für das 2. Steuerregister (VIC+22). Normal: 200,200,192
ML: $C55F/$C560/$C561

**MC n1,n2,n3**

Dasselbe für das Memory-Control-Register (VIC+24). Normal: 21,52,52
ML: $C562/$C563/$C564

**CI n1,n2,n3**

Das Gleiche gilt für die Basisadresse der CIA 2 (56576).
Normal: 151,148,148
ML: $C565/$C566/$C567

**RZ z1,z2,z3**

Legt die Rasterzeilen für die drei Interrupts fest. Normal: 81,88,248 andere Möglichkeit: 106,115,232 + SU 4
ML: $C56E/$C56F/$C570

**SU n**

Behebt Fehler, die bei der Änderung der Rasterzeilen entstehen können. Das erkennt man am Rucken des Scrollings.
n kann Werte von 0 bis 7 annehmen.
Den richtigen Wert kann man nur durch Ausprobieren feststellen.
Wenig Probleme wird man haben, wenn man die RZ-Werte in Achterschritten verändert. Normal: n=5
ML: n:$C571
Die folgenden zwei Befehle sind als allgemeine Hilfe gedacht:

**FU a,e,n,w**

Dieser Befehl füllt den Speicher von a bis e mit n, während in der Speicherstelle 1 w steht. w kann weggelassen werden.
ML: a:$FB/$FC, e:$FD/$FE, n:$03B0, LDX #w : JSR $C842

**UT a,e,a2,r,w**

Es wird der Block von a bis e nach a2 übertragen. Dabei steht in 1 während des Lesens r und während des Schreibens w.
ML: a:$FB/$FC, e:$03A6/$03A7, a2:$FD/$FE,
LDA #r : LDX #w : JSR $C725
Natürlich kann man die Erweiterung auch ausschalten:

**OF**

Schaltet die Erweiterung vollständig ab. Sie muß mit SYS49152 wieder gestartet werden. Die Wirkung ähnelt der der Run-Stop/Restore-Funktion.
Dieser Befehl wird zum Beispiel dann benötigt, wenn man etwas von der Datasette laden will.
ML: JSR $C65E

## Die Diskettenbefehle

**LD "Name"**

Lädt den Pseudoschirm mit der Bezeichnung Name von der Diskette. Die Anfangsadresse, die X-Ausdehnung und die Y-Ausdehnung werden mitgeladen und gesetzt.
ML: LDA #0 : STA $CBF8
LDX #<Namenadr. : LDY #>Namenadr. : LDA #Namenlänge
JSR $FFBD (Setnam-Routine)
JSR $CAC8

**LD "Name",aa**

Lädt ein beliebiges Datenfile absolut an die Adresse aa von der Diskette. Achtung: Das File wird vollständig geladen, inklusive den beiden Adreßbytes am Anfang jedes Programms. Man sollte deshalb von aa zwei abziehen.
Wenn man sequentielle Files oder Userfiles laden will, dann hängt man an den Namen einfach noch ,S oder ,U an.
ML: aa:$FB/$FC, LDA #1 : STA $CBF8
Ansonsten weiter wie oben.

**SV "Name",ea**

Speichert einen Pseudoschirm mit der Bezeichnung »Name« auf Diskette. Die Endadresse ea errechnet sich aus Anfangsadresse + X-Ausdehnung * Y-Ausdehnung.
ML: ea:$FD/$FE
LDX #<Namenadr. : LDY #>Namenadr. : LDA #Namenlänge
JSR $FFBD (Setnam-Routine)
JSR $CB3E

## Die USR(x)-Funktion

Mit ihrer Hilfe kann sich der Benutzer einige wichtige Werte holen. Je nachdem, welchen Wert x annimmt, wird zu verschiedenen Routinen verzweigt.

**x=0** Jedesmal, wenn eine Grenze berührt wird, die mit dem GR-Befehl gesetzt wurde, setzt das Programm ein Flag auf eins. Mit dieser Funktion wird das Flag abgefragt und gelöscht.
ML: $03AD

**x=1** Liefert die momentane X-Position.
ML: $0358/$0359

**x=2** Liefert die Y-Position.
ML: $035A/$035B

**x=3** Gibt die Adresse im Pseudoschirm an, die der Bildmitte entspricht.
ML: $C50C/$C50D

**x=4** Ergibt eins, wenn der Feuerknopf des Joysticks in Port 2 gedrückt ist.
ML: Es empfiehlt sich der normale Joystickport $DC00.

**x=5** Dasselbe für Port 1.
ML: $DC01

**x=10** Übermittelt die Anfangsadresse des Pseudoschirms (die möglicherweise nicht bekannt ist, weil der Schirm eben geladen wurde).
ML: $CBFA/$CBFB

**x=11** Liefert die X-Ausdehnung.
ML: $CBFC/$CBFD

**x=12** Liefert die Y-Ausdehnung.
ML: $CBFE/$CBFF

**x>255** Übermittelt den Inhalt der Speicherstelle x. Es wird immer der Inhalt des RAMs gelesen. Man hat damit also die Möglichkeit, das versteckte RAM zu lesen.
ML: Die allgemeine Methode funktioniert so:
SEI: LDX #48:STX 1:LDA x:LDX #55:STX 1:CLI

Übrigens: In $036A/$036B (874/875) und $036C/$036D (876/877) stehen die X- und Y-Werte für die Trägheitsgeschwindigkeit.

## Der Pseudoschirmeditor

Der Editor ist ein Basic-Programm, das mit den wichtigsten Funktionen zur komfortablen Erstellung eines Pseudoschirms ausgestattet ist.

Er benötigt die Scroll-Machine, um arbeiten zu können. Man muß sie also von Datasette oder Diskette laden. Dann lädt man den Editor und startet ihn mit RUN.

Zuerst wird man nach der X/Y-Ausdehnung und der Anfangsadresse gefragt. Die Werte XD=128, YD=64 und AA=40960 sind vorgegeben, so daß der Pseudoschirm das RAM unter dem Basic-ROM voll belegt ($A000 — $BFFF). In diesem Fall ist RETURN zu drücken. Dann will das Programm wissen, ob der Zeichensatz vom ROM ins RAM kopiert werden soll. Wenn man das Programm eben erst geladen hat, sollte man mit »J« antworten. Steht jedoch schon ein neuer Zeichensatz, der erhalten werden soll, im Speicher, so gibt man »N« ein. Anschließend erscheint die Frage »MULTICOLOR (J/N)« auf dem Bildschirm. Wenn der normale Commodore-Zeichensatz benutzt wird, lautet die Antwort immer »N«, ansonsten hängt das vom Zeichensatz ab. Natürlich muß man auch noch die Farben eingeben, wenn man nur Return drückt, erhält man die Standardeinstellung. Schließlich kann man entscheiden, ob der Pseudoschirm gelöscht werden soll. Normalerweise antwortet man mit »J« und der Pseudoschirm wird nach einer Rückversicherung mit dem Zeichen 32 (Space) gefüllt.

Nach diesen Eingaben hat man endlich das normale Bild des Editors vor sich, das folgendermaßen aufgebaut ist:

Der Bildschirm ist in zwei Bereiche unterteilt. Oben wird immer der Teil des Pseudoschirms gezeigt, der gerade bearbeitet wird. In der Mitte dieses Bereiches befindet sich ein Sprite, der als Cursor fungiert. Der Bildschirm läßt sich mit Hilfe eines Joysticks in Port zwei bewegen. Unten kann man den kompletten C 64-Zeichensatz erkennen. Er ist in acht Zeilen à 32 Zeichen unterteilt, wie bei vielen Zeichengeneratorprogrammen. Es wird der große Zeichensatz benützt, weil er mehr Grafikzeichen zur Verfügung stellt. Auch hier wird ein Sprite als Cursor benutzt, der sich mit den Cursortasten auf der Zeichenpalette bewegen läßt.

Im Normalmodus wird durch einen Druck auf den Joystickknopf jeweils das Zeichen von der Palette auf den Pseudoschirm übertragen, auf das der Zeichencursor zeigt.

Das ist die einfachste Funktion des Programms, mit ihr allein ist es jedoch fast unmöglich, einen Pseudoschirm zu erstellen. Deshalb bietet es noch einige andere Möglichkeiten wie Rechteck füllen, Rechteck übertragen und so weiter, die man von Programmen wie Paint Magic oder Koalapainter her kennt. Sie bedürfen keiner langen Erklärung, da die Bedienung einfach und logisch aufgebaut ist.

Auch der Mikromodus ist schnell erklärt. Er benutzt die Zeichen 32, 97, 98, 108, 123, 124, 126, 127 und ihre reversen Entsprechungen, um eine vierfache Auflösung zu erreichen. Mit dem Klammeraffen kann man zwischen Punkt löschen und Punkt setzen umschalten. Mit Shift-M kann man die Zeichenfolge neu definieren. Wenn man einen selbstdefinierten Zeichensatz benutzt, stimmen die Codes 97, 98, 108 etc. nicht mehr. Der Einfachheit halber müssen die neuen Zeichen aufeinanderfolgen, mit Leerraum (=Space) beginnen und mit einem vollständig gefüllten Zeichen enden. Eine nicht ganz einfache Funktion ist der Konturmodus, der im folgenden näher erläutert wird.

Mit den Zeichen 100, 111, 121, 98 und so weiter, kann man eine sanft aufsteigende Linie bilden. Das wird im Konturmodus ausgenutzt. Wenn man F7 betätigt, wird dieser Modus gesetzt. Jetzt kann man Landschaften formen, die sanft auf- und absteigen. Jedoch kann man keine Höhlendecken oder ähnliches bilden, auch senkrechte Wände bleiben einem versagt. Das kann man mit einem Druck auf »K« ändern. Man wird nach X+, X—, Y+, Y— gefragt. Mit den Tasten 1-4 kann man die gewünschte Einstellung wählen. X bedeutet senkrechte Wände, Y steht für waagrechte Wände. Plus und Minus geben jeweils an, welche Seite Freiraum und welche Wand ist. Anschließend muß man noch die Geschwindigkeit angeben, die regelt, ob der Berg sanft oder steil ansteigt.

Wenn man senkrechte Wände bearbeitet, wird man feststellen, daß die Kontur fehlerhaft ist. Das liegt am Zeichensatz des C 64. Die Zeichen 101 und 116 sind identisch, was von Commodore so geregelt wurde, damit das Zeichen auf einem Fernsehschirm besser sichtbar ist. Mit einem VC 20-Zeichensatz hätten wir diese Probleme nicht. Wenn man einen eigenen Zeichensatz benutzt, ist das Problem auch gelöst. Denn wie beim Mikromodus gibt es hier ebenfalls die Möglichkeit, das Programm an den eigenen Zeichensatz anzupassen.

Mit SHIFT-K ruft man die entsprechende Routine auf. Man benötigt vier Zeichenfolgen. Jeweils eine für X+, X—, Y+, Y—. Mit 1-4 kann man wählen, welche man neu definieren will. Jede Zeichenfolge muß aus acht Zeichen bestehen, von denen das erste Space (32) oder das letzte Revers-Space (160) ist. Man steuert mit den Cursortasten auf das erste Zeichen und drückt dann RETURN. Nun muß man sich für +1 oder —1 entscheiden. Hier hilft nur ausprobieren. Dann gibt man nur noch die Geschwindigkeit an und die neue Zeichenfolge ist im Speicher.

Ein Pseudoschirm muß auch abgespeichert und geladen werden können, was man mit F1 und F3 erreichen kann. Die Scroll-Machine enthält bereits Routinen, um einen Pseudoschirm aus irgendeinem Speicherbereich auf Diskette Geräte-

nummer 8 abzuspeichern und in den gleichen Bereich wieder zu laden. Für die Datasette existiert jedoch kein Befehl, so daß man dafür die ROM-Routinen verwenden muß. Deshalb kann man mit der Datasette nur das Basic-RAM benutzen, da sich die Bereiche von 40960 bis 49151 und 57344 bis 65535 von den ROM-Routinen nicht auslesen lassen. Man kann jedoch einen Pseudoschirm in das Basic-RAM laden, um ihn dann mit dem UT-Befehl in das versteckte RAM zu schieben. Wenn man einen Pseudoschirm mit dem eigenen Programm von Kassette laden will, muß man das Unterprogramm des Editors verwenden (ab Zeile 17 500). Der Wechselmodus wird mit »W« aktiviert und mit SHIFT-W definiert. Man kann einen Bereich von Zeichen festlegen, die dann zyklisch wiederholt verwendet werden. Nützlich ist dies für flächige Felsstrukturen oder ähnliches. Denn wenn man den ganzen Bereich mit dem gleichen Zeichen gestaltet, wird die Struktur regelmäßig und sieht nicht echt aus.

Aus allen Modi kommt man wieder heraus, indem man den Pfeil nach links drückt. Dadurch wird der Editor wieder in den Normalzustand zurückgesetzt.

Zum Editor ist damit alles Wichtige gesagt. Abtippen muß man ihn wie jedes andere Programm, die REM- und Doppelpunkt-Zeilen werden jedoch nicht angesprungen, deshalb können sie weggelassen werden.

Die Befehlsliste gibt Auskunft über die restlichen Funktionen und deren Handhabung.

Ich hoffe, daß das Programm möglichst fehlerlos ist und wünsche Ihnen viel Spaß damit.

Um die Fähigkeiten von »Scroll-Machine« aufzuzeigen, soll auch gleich ein kleines Beispielprogramm angeführt werden. Da man es niemandem zumuten kann, einen Pseudoschirm von 8 KByte oder mehr abzutippen, muß sich das Spiel den Schirm selbst aufbauen. Daß darunter die Qualität der Grafik leidet ist klar. Außerdem habe ich keinen neudefinierten Zeichensatz verwendet, sondern nur die Commodore-Zeichen Space (32), Invers-Space (160) und Invers-Shift-0 (207). Der Pseudoschirm besitzt nur eine Auflösung von 128 x 64, was genau 8 KByte entspricht. Eigene Schirme sollte man größer machen — etwa 20 bis 30 KByte — und wenn möglich mit einem eigenen Zeichensatz versehen.

## Atomdepot, ein Demoprogramm zur Scroll-Machine

Doch dieses Spiel ist trotzdem ein gutes Beispiel dafür, wie man die Scroll-Machine benutzt.

Der Handlungsort ist ein Bergwerksstollen in nicht allzu ferner Zukunft, in dem noch drei Atommüllfässer lagern. Sinn des Spiels ist es, die drei Fässer von der hinteren Kammer in die vordere zu bringen, von wo sie abtransportiert werden können. Der Greifer des Transporters wird mit dem Feuerknopf des Joysticks (Port 2) bedient, doch erst dann, wenn Greifarme punktgenau am Faß anliegen. Ebenso vorsichtig muß man es auf den Boden stellen. In einem engen Stollen ist der Transport nicht einfach, doch mit genügend Geduld und etwas Übung ist auch das kein Problem. (Thilo Herrmann/ah)

| Taste | Funktion |
|---|---|
| **Ziffern 1 bis 9** | Regeln die Geschwindigkeit des Cursors. Die Geschwindigkeit Eins ist normal. |
| **Space** | Auf den Schnellgang schalten. |
| **Pfeil links** | Den Editor auf den Normalmodus setzen. |
| **CRSR-Tasten** | Mit ihnen kann man den Zeichencursor auf der Palette bewegen, um ein Zeichen auszuwählen. |
| **DEL** | Löschen des Zeichens unter dem Cursor. |
| **HOME** | Setzen des Zeichencursors auf das Leerzeichen. |
| **L** | Löschen des Pseudoschirms. |
| **F** | Farben neu bestimmen. |
| **C** | Zeichensatz kopieren. |
| **Z** | Zeichensatz laden. |
| **F1** | Abspeichern eines Pseudoschirms auf Disk oder Datasette. |
| **F3** | Laden eines Pseudoschirms von Disk oder Datasette. |
| **M** | Mikromodus einschalten. |
| **SHIFT-M** | Mikromodus neu definieren. |
| **Klammeraffe** | Umschalten des Mikromodes zwischen Punkt setzen und Punkt löschen. |
| **F7** | Konturmodus einschalten. |
| **K** | Kontur und Geschwindigkeit wählen. |
| **SHIFT-K** | Kontur neu definieren. Das erste Zeichen wird mit den Cursortasten ausgewählt und mit Return festgelegt. |
| **Pfeil oben** | Kontur- und Mikromodus zurücksetzen. |
| **W** | Wechselmodus einschalten. |
| **SHIFT-W** | Wechselnde Zeichen neu definieren. Die Zeichen kann man mit den Cursortasten auswählen und mit Return festlegen. |
| **U** | Mehrere zusammenhängende Zeichen von der Palette übertragen. Exit mit Return. |
| **S** | Mit der Tastatur direkt auf den Schirm nach Art einer Schreibmaschine schreiben. Steuertasten: RVS ON, RVS OFF, DEL, Cursortasten, RETURN. Exit mit F7. |
| **R** | Rechteck mit einem beliebigen Zeichen füllen. Der Punkt an dem sich der Cursor gerade befindet, wird nun als Eckpunkt betrachtet. Den gegenüberliegenden Eckpunkt wählt man mit dem Joystick und abschließendem Knopfdruck. Das Zeichen, mit dem das Rechteck ausgefüllt wird, ist durch den Zeichencursor auf der Palette festgelegt. Benutzung des Schnellgangs ist möglich. Abbruch mit Pfeil links. |
| **T** | Rechteck übertragen. Der momentane Ort des Cursors wird als Ecke des Rechtecks angesehen. Die diagonal gegenüberliegende Ecke wird durch Bewegung des Cursors mit dem Joystick an die entsprechende Stelle festgelegt. Anschließend legt man auf die gleiche Weise die linke obere Ecke der Kopie fest. Eine Benutzung des Schnellgangs ist ebenfalls möglich. Abbruch mit Pfeil links. |
| **A** | Ändern der X/Y-Ausdehnung und der Anfangsadresse. Wenn man auf die der Eingabe folgende Frage mit »J« antwortet, wird der komplette Pseudoschirm an die neue Anfangsadresse kopiert und vollständig umformatiert. Die dafür benötigte Zeit hängt maßgeblich von der Y-Ausdehnung ab. |
| **CTRL-X** | Verlassen des Programms. |

**Tabelle 1. Zusammenfassung der Editorbefehle**

```
programm : scroll-machine  c000 cc00
--------------------------------------

c000 : a9 72 8d 08 03 a9 c5 8d  f6
c008 : 09 03 a9 1a 8d 11 03 a9  01
c010 : ca 8d 12 03 a2 00 bd c2  2c
c018 : c9 49 d3 20 d2 ff e8 e0  11
c020 : 2a d0 f3 60 78 a9 7f 8d  a9
c028 : 0d dc a9 01 8d 1a d0 ad  76
c030 : 6e c5 8d 12 d0 a9 18 8d  fc
c038 : 11 d0 a9 00 85 02 8d ff  ba
c040 : ff a9 4d 8d 14 03 a9 c0  9a
c048 : 8d 15 03 58 60 ad 19 d0  a5
c050 : 8d 19 d0 a5 02 f0 0e c9  c6
c058 : 01 f0 49 c9 02 d0 03 4c  a8
c060 : 84 c4 4c 2d c2 ad 69 c5  ca
c068 : 8d 21 d0 ad 5d 03 8d 11  b6
c070 : d0 ad 66 c5 8d 00 dd ad  15
c078 : 6c c5 8d 20 d0 ad 60 c5  b6
c080 : 8d 16 d0 ad 63 c5 8d 18  cd
c088 : d0 ad 6f c5 8d 12 d0 e6  3e
c090 : 02 a9 ec 8d 18 03 a0 c9  03
c098 : 8c 19 03 8d fa ff 8c fb  fd
c0a0 : ff 4c cd c4 ad 5c 03 8d  b6
c0a8 : 16 d0 ad 6a c5 8d 21 d0  ce
c0b0 : ad 64 c5 8d 18 d0 ad 6d  4c
c0b8 : c5 8d 20 d0 ad 67 c5 8d  ae
c0c0 : 00 dd ad 70 c5 8d 12 d0  db
c0c8 : e6 02 ad b3 03 d0 0e ad  db
c0d0 : b4 03 8d b3 03 ad 95 03  da
c0d8 : f0 0b ce 95 03 a9 01 8d  51
c0e0 : 64 03 4c 2d c2 ad 94 03  70
c0e8 : 8d 95 03 a9 ff 38 ed 58  60
c0f0 : 03 29 07 18 6d 61 c5 8d  61
c0f8 : 5c 03 ad 5a 03 18 6d 71  16
c100 : c5 8d 5e 03 a9 ff 38 ed  db
c108 : 5e 03 29 07 18 6d 5e c5  05
c110 : 8d 5d 03 a2 03 bd 58 03  e6
c118 : 9d 52 03 ca 10 f7 a2 03  4a
c120 : 4e 53 03 6e 52 03 4e 55  c8
c128 : 03 6e 54 03 ca d0 f1 ad  2e
c130 : a2 03 f0 03 4c 4f c1 a2  7c
c138 : 03 bd 52 03 dd 60 03 9d  37
c140 : 60 03 d0 0b ca 10 f2 a9  04
c148 : 01 8d 64 03 4c c2 c1 ac  c5
c150 : 54 03 20 0a ca b9 00 de  a7
c158 : 8d d6 c4 b9 00 df ac 55  15
c160 : 03 18 79 00 de 8d d7 c4  11
c168 : 20 14 ca ad d6 c4 18 6d  c9
c170 : 52 03 8d d6 c4 ad d7 c4  25
c178 : 6d 53 03 8d d7 c4 ad d6  09
c180 : c4 18 6d fa cb 8d d6 c4  19
c188 : ad d7 c4 6d fb cb 8d d7  04
c190 : c4 a2 01 a9 00 8d b9 03  74
c198 : a0 07 bd d5 c4 18 6d fc  a3
c1a0 : cb 99 d5 c4 bd d6 c4 6d  c6
c1a8 : fd cb 99 d6 c4 8a 18 69  a0
c1b0 : 06 aa 98 18 69 06 a8 ee  7c
c1b8 : b9 03 ad b9 03 cd b7 03  19
c1c0 : 90 d8 ad 3e 03 d0 03 4c  4b
c1c8 : 22 c2 a0 01 b9 74 03 18  0f
c1d0 : 79 72 03 aa 29 07 99 72  af
c1d8 : 03 8a 4a 4a 4a 99 70 03  35
c1e0 : 88 10 e9 ae a5 03 bd 00  2a
c1e8 : dc a2 01 a0 02 8d ac 03  af
c1f0 : 29 01 d0 03 20 30 c2 ad  18
c1f8 : ac 03 29 02 d0 03 20 5c  0f
c200 : c2 ad ac 03 ca 88 88 29  8a
c208 : 04 d0 03 20 30 c2 ad ac  62
c210 : 03 29 08 d0 03 20 5c c2  ec
c218 : ad 3e 03 c9 01 f0 03 20  c2
c220 : 81 c2 20 9a c3 ad 9e 03  88
c228 : f0 03 20 04 c4 4c 31 ea  6c
c230 : b9 58 03 38 fd 70 03 99  80
c238 : 58 03 b9 59 03 e9 00 99  5e
c240 : 59 03 b0 05 a9 01 99 9f  30
c248 : 03 b9 6a 03 38 fd 4a 03  c5
c250 : 99 6a 03 b9 6b 03 e9 00  8d
c258 : 99 6b 03 60 b9 58 03 18  0e
c260 : 7d 70 03 99 58 03 b9 59  41
c268 : 03 69 00 99 59 03 b9 6a  bd
c270 : 03 18 7d 4a 03 99 6a 03  d5
c278 : b9 6b 03 69 00 99 6b 03  55
c280 : 60 ae 6a 03 ac 6b 03 20  a5
c288 : d7 c2 8e 7a 03 8c 7b 03  3c
c290 : 8d 48 03 ae 6c 03 ac 6d  44
c298 : 03 20 d7 c2 8e 7c 03 8c  eb
c2a0 : 7d 03 8d 49 03 20 2b c3  91
c2a8 : a2 01 a0 02 b9 7a 03 18  df
c2b0 : 7d 4e 03 8d ac 03 b9 7b  88
c2b8 : 03 69 00 99 7b 03 ad ac  83
c2c0 : 03 9d 4e 03 b9 7b 03 99  3d
c2c8 : 7a 03 a9 00 99 7b 03 88  c1
c2d0 : 88 ca 10 d8 4c ee c2 98  55
c2d8 : c9 80 90 0f 98 49 ff a8  0c
c2e0 : 8a 49 ff aa e8 d0 01 c8  0f
c2e8 : a9 ff 60 a9 01 60 a2 01  7e
c2f0 : a0 02 bd 48 03 30 15 b9  83
c2f8 : 58 03 18 79 7a 03 99 58  de
c300 : 03 b9 59 03 69 00 99 59  46
c308 : 03 4c 25 c3 b9 58 03 38  ce
c310 : f9 7a 03 99 58 03 b9 59  72
c318 : 03 e9 00 99 59 03 b0 05  be
c320 : a9 01 99 9f 03 88 88 ca  d0
c328 : 10 c8 60 a0 02 a2 01 b9  75
c330 : 8f 03 d9 7b 03 90 0a d0  a5
c338 : 14 b9 8e 03 d9 7a 03 b0  0c
c340 : 0c b9 8e 03 99 7a 03 b9  1a
c348 : 8f 03 99 7b 03 b9 7a 03  1d
c350 : 38 fd 92 03 99 7a 03 b9  79
c358 : 7b 03 e9 00 99 7b 03 10  71
c360 : 08 a9 00 99 7a 03 99 7b  8d
c368 : 03 88 88 ca 10 c1 ae 7a  e9
c370 : 03 ac 7b 03 ad 48 03 10  52
c378 : 03 20 dc c2 8e 6a 03 8c  7c
c380 : 6b 03 ad 6a 03 ae 7c 03  c3
c388 : ac 7d 03 ad 49 03 10 03  5c
c390 : 20 dc c2 8e 6c 03 8c 6d  8d
c398 : 03 60 a2 00 bd 85 03 20  c8
c3a0 : f7 c3 9d 89 03 98 9d 8b  94
c3a8 : 03 bd 89 03 7d 87 03 a8  be
c3b0 : 29 07 9d 87 03 98 4a 4a  68
c3b8 : 4a 8d ff 03 8a 0a a8 bd  40
c3c0 : 8b 03 30 15 ad ff 03 18  93
c3c8 : 79 58 03 99 58 03 b9 59  99
c3d0 : 03 69 00 99 59 03 4c ee  78
c3d8 : c3 b9 58 03 38 ed ff 03  e7
c3e0 : 99 58 03 b9 59 03 e9 00  f3
c3e8 : 99 59 03 8c c0 5d e8 e0  dd
c3f0 : 02 b0 03 4c 9c c3 60 a0  3f
c3f8 : 01 c9 80 90 06 49 ff a8  0c
c400 : 98 a0 ff 60 a2 00 a0 00  a1
c408 : bd 9f 03 d0 09 20 67 c4  28
c410 : f0 07 c9 01 f0 03 20 38  2e
c418 : c4 c8 c8 20 67 c4 f0 07  e5
c420 : c9 ff f0 03 20 38 c4 c8  ee
c428 : c8 e8 e8 e0 04 90 d9 a9  3a
c430 : 00 8d 9f 03 8d a1 03 60  f2
c438 : a9 01 8d ad 03 ad 9e 03  99
c440 : c9 01 f0 16 98 4a 29 01  0b
c448 : d0 08 c8 c8 20 5a c4 88  60
c450 : 88 60 88 88 20 5a c4 c8  b5
c458 : c8 60 b9 96 03 9d 58 03  16
c460 : b9 97 03 9d 59 03 60 b9  fc
c468 : 97 03 dd 59 03 90 0f d0  b6
c470 : 0a b9 96 03 dd 58 03 90  2b
c478 : 05 f0 06 a9 ff 60 a9 01  58
c480 : 60 a9 00 60 ad 6b c5 8d  29
c488 : 20 d0 ad 5c c5 8d 11 d0  b6
c490 : ad 62 c5 8d 18 d0 ad 65  1b
c498 : c5 8d 00 dd ad 6e c5 8d  60
c4a0 : 12 d0 a9 00 85 02 ad 5f  63
c4a8 : c5 8d 16 d0 ad 68 c5 8d  24
c4b0 : 21 d0 ad 64 03 d0 11 a5  77
c4b8 : 01 8d af 03 a9 30 85 01  00
c4c0 : 20 d3 c4 ad af 03 85 01  dc
c4c8 : a9 00 8d 64 03 68 a8 68  48
c4d0 : aa 68 40 a2 00 bd a1 04  8f
c4d8 : 9d a0 04 bd c9 04 9d c8  43
c4e0 : 04 bd f1 04 9d f0 04 bd  ad
c4e8 : 19 05 9d 18 05 bd 41 05  3c
c4f0 : 9d 40 05 bd 69 05 9d 68  ac
c4f8 : 05 bd 91 05 9d 90 05 bd  cf
c500 : b9 05 9d b8 05 bd e1 05  8a
c508 : 9d e0 05 bd 09 06 9d 08  56
c510 : 06 bd 31 06 9d 30 06 bd  f1
c518 : 59 06 9d 58 06 bd 81 06  47
c520 : 9d 80 06 bd a9 06 9d a8  c9
c528 : 06 bd d1 06 9d d0 06 bd  36
c530 : f9 06 9d f8 06 bd 21 07  94
c538 : 9d 20 07 bd 49 07 9d 48  33
c540 : 07 bd 71 07 9d 70 07 bd  58
c548 : 99 07 9d 98 07 bd c1 07  53
c550 : 9d c0 07 e8 e0 28 f0 03  45
c558 : 4c d5 c4 60 1b 17 10 c8  08
c560 : c8 c0 15 34 34 97 94 94  d0
c568 : 06 01 0e 06 01 06 51 58  69
c570 : f8 05 20 73 00 c9 5f f0  0f
c578 : 06 20 79 00 4c e7 a7 20  d0
c580 : 85 c5 4c ae a7 a0 01 a2  99
c588 : 00 b1 7a dd 86 c9 d0 0b  cb
c590 : c8 e8 b1 7a dd 86 c9 f0  a3
c598 : 0e ca 88 e8 e8 e0 3c b0  32
c5a0 : 03 4c 89 c5 4c 08 af 20  e8
c5a8 : 73 00 ca bd 4a c9 8d be  2c
c5b0 : c5 bd 4b c9 8d bf c5 20  8e
c5b8 : 73 00 20 73 00 4c 5e c6  0b
c5c0 : a2 a8 a9 00 9d 3b 03 ca  76
c5c8 : d0 fa 8d fa cb 8d fd cb  91
c5d0 : a9 a0 a2 cc 8d d9 c4 8e  e3
c5d8 : da c4 a2 15 20 ae c7 a9  49
c5e0 : a0 8d fb cb a9 80 8d fc  8e
c5e8 : cb 20 ac c6 a9 fe 8d 61  53
c5f0 : 03 8d 63 03 60 20 9e b7  e4
c5f8 : 8a d0 2f a9 ff 85 02 ad  7b
c600 : 68 c5 8d 21 d0 ad 6b c5  86
c608 : 8d 20 d0 ad 5c c5 29 f8  1a
c610 : 09 0b 8d 11 d0 ad 65 c5  c0
c618 : 8d 00 dd ad 5f c5 09 08  2b
c620 : 8d 16 d0 ad 62 c5 8d 18  5d
c628 : d0 60 20 24 c0 60 20 3f  c3
c630 : c9 8d 58 03 8e 59 03 20  36
c638 : fd ae 20 3f c9 8d 5a 03  f5
c640 : 8e 5b 03 60 20 3f c9 8d  87
c648 : 85 03 20 fd ae 20 3f c9  93
c650 : 8d 86 03 60 20 04 c9 8d  52
c658 : 4a 03 8e 4b 03 60 78 a9  99
c660 : e4 8d 08 03 a9 a7 8d 09  8d
c668 : 03 20 15 fd 20 a3 fd 20  d8
c670 : 18 e5 58 60 20 04 c9 8d  01
c678 : 92 03 8e 93 03 60 20 26  a2
c680 : c9 8d 8e 03 8e 8f 03 20  c6
c688 : 38 c9 8d 90 03 8e 91 03  0b
c690 : 60 20 26 c9 8d fa cb 8e  c0
c698 : fb cb 20 38 c9 8d fc cb  1c
c6a0 : 8e fd cb 20 38 c9 8d fe  2a
c6a8 : cb 8e ff cb 78 20 0a ca  7a
c6b0 : a2 00 86 fb 86 fc a5 fb  52
c6b8 : 9d 00 de a5 fc 9d 00 df  3e
c6c0 : a5 fb 18 6d fc cb 85 fb  53
c6c8 : a5 fc 6d fd cb 85 fc e8  b5
c6d0 : d0 e4 20 14 ca 58 60 a9  e1
c6d8 : 00 a2 04 9d 69 03 ca d0  5a
c6e0 : fa 20 9e b7 8e 3e 03 f0  52
c6e8 : 17 20 31 c9 8e 74 03 20  6e
c6f0 : 31 c9 8e 75 03 f0 09 20  74
c6f8 : 31 c9 8a 29 01 8d a5 03  ef
c700 : 60 20 9e b7 8e 9e 03 60  b9
c708 : 20 26 c9 85 fb 86 fc 20  86
c710 : 38 c9 8d a6 03 8e a7 03  ae
c718 : 20 38 c9 85 fd 86 fe 20  c8
c720 : fd ae 20 04 c9 8d a9 03  b3
c728 : 8e aa 03 a5 01 8d a8 03  a6
c730 : a0 00 78 ae a9 03 86 01  93
c738 : b1 fb ae aa 03 86 01 91  73
c740 : fd a5 fc cd a7 03 90 0e  fa
c748 : a5 fb cd a6 03 90 07 ad  5f
c750 : a8 03 85 01 58 60 e6 fb  17
c758 : d0 02 e6 fc e6 fd d0 02  28
c760 : e6 fe 4c 33 c7 20 26 c9  e8
c768 : 8d 96 03 8e 97 03 20 38  55
c770 : c9 8d 98 03 8e 99 03 20  88
c778 : 38 c9 8d 9a 03 8e 9b 03  65
c780 : 20 38 c9 8d 9c 03 8e 9d  38
c788 : 03 60 20 9e b7 8e 95 03  e3
c790 : 8e 94 03 60 20 9e b7 8e  28
c798 : a2 03 60 20 26 c9 8d d9  73
c7a0 : c4 8e da c4 20 31 c9 e0  6f
c7a8 : 16 90 03 4c 48 b2 8e b7  14
c7b0 : 03 78 a9 00 8d b8 03 a2  4a
c7b8 : 04 a0 0a a9 9d 9d d4 c4  68
c7c0 : a9 bd 9d d1 c4 ee b8 03  96
c7c8 : ad b8 03 cd b7 03 b0 1e  de
c7d0 : bd d5 c4 18 69 28 99 d5  96
c7d8 : c4 bd d6 c4 69 00 99 d6  74
c7e0 : c4 8a 18 69 06 aa 98 18  65
c7e8 : 69 06 a8 4c bb c7 a0 7c  7d
c7f0 : 8a 69 09 8d ba 03 b9 d7  7d
c7f8 : c4 9d d7 c4 e8 c8 ec ba  17
c800 : 03 90 f3 58 ce b7 03 60  cb
c808 : 20 11 c9 8d 5c c5 8e 5d  be
c810 : c5 8c 5e c5 60 20 11 c9  4a
c818 : 8d 5f c5 8e 60 c5 8c 61  c1
c820 : c5 60 20 26 c9 85 fb 86  a8
c828 : fc 20 38 c9 8d b1 03 8e  0b
c830 : b2 03 a5 01 8d ae 03 20  88
c838 : 31 c9 8e b0 03 f0 0b 20  2c
c840 : 31 c9 78 a5 01 8d ae 03  66
c848 : 86 01 ad b0 03 a0 00 91  29
c850 : fb a6 fb ec b1 03 d0 0e  cd
c858 : a6 fc ec b2 03 90 07 ad  3a
c860 : ae 03 85 01 58 60 e6 fb  2d
c868 : d0 02 e6 fc 4c 4f c8 20  35
c870 : 11 c9 8d 62 c5 8e 63 c5  ff
c878 : 8c 64 c5 60 20 11 c9 8d  81
c880 : 68 c5 8e 69 c5 8c 6a c5  92
```

**Listing 1. Hauptprogramm »Scroll-Machine«. Das Programm ist mit dem MSE einzugeben.**

```
c888 : 60 20 9e b7 e0 01 d0 09  02
c890 : 8e b4 03 a9 00 8d b3 03  b0
c898 : 60 a9 00 8d b4 03 8d b3  80
c8a0 : 03 60 20 11 c9 8d 6b c5  40
c8a8 : 8e 6c c5 8c 6d c5 60 20  36
c8b0 : 11 c9 8d 6e c5 8e 6f c5  f1
c8b8 : 8c 70 c5 60 a9 00 a2 d0  c0
c8c0 : 85 fb 86 fc 85 fd 86 fe  e4
c8c8 : a9 ff a2 d7 8d a6 03 8e  4c
c8d0 : a7 03 a9 33 a2 30 4c 25  f1
c8d8 : c7 60 20 11 c9 8d 65 c5  24
c8e0 : 8e 66 c5 8c 67 c5 60 20  0b
c8e8 : 9e b7 8a a2 00 9d 00 d8  f7
c8f0 : 9d 00 d9 9d 00 da 9d 00  05
c8f8 : db e8 d0 f1 60 20 9e b7  ab
c900 : 8e 71 c5 60 20 9e b7 8e  b7
c908 : ab 03 20 31 c9 ad ab 03  22
c910 : 60 20 04 c9 8d a3 03 8e  da
c918 : a4 03 20 31 c9 8a a8 ad  5b
c920 : a3 03 ae a4 03 60 20 8a  4e
c928 : ad 20 f7 b7 a5 14 a6 15  9a
c930 : 60 20 fd ae 20 9e b7 60  8c
c938 : 20 fd ae 20 26 c9 60 20  79
c940 : 8a ad 20 9b bc a5 65 a6  f8
c948 : 64 60 89 c8 c0 c5 f5 c5  f5
c950 : 2e c6 44 c6 54 c6 74 c6  a6
c958 : 7e c6 d7 c6 01 c7 65 c7  7b
c960 : 8a c7 94 c7 08 c8 15 c8  99
c968 : 6f c8 7c c8 a2 c8 af c8  34
c970 : e7 c8 fd c8 da c8 9b c7  46
c978 : 08 c7 91 c6 22 c8 b4 ca  72
c980 : 32 cb 5e c6 bc c8 53 59  1a
c988 : 52 53 55 53 50 4f 42 57  7b
c990 : 54 42 54 52 54 47 4a 53  b4
c998 : 47 4d 47 52 41 53 54 4d  3d
c9a0 : 31 52 32 52 4d 43 48 46  6e
c9a8 : 52 46 52 5a 5a 46 53 55  cd
```

```
c9b0 : 43 49 56 49 55 54 4d 54  2c
c9b8 : 46 55 4c 44 53 56 4f 46  f6
c9c0 : 43 50 de c1 f3 80 90 81  a4
c9c8 : 9c 9f 9f fe 9e 92 90 9b  f3
c9d0 : 9a 9d 96 ff f3 e2 ea eb  b8
c9d8 : e6 f3 91 8a f3 87 9b 9a  8d
c9e0 : 9f 9c f3 9b 96 81 81 9e  f6
c9e8 : 92 9d 9d f3 a9 37 85 01  9b
c9f0 : a2 00 a0 00 8e 11 d0 e8  41
c9f8 : d0 fd c8 d0 fa a9 04 8d  3b
ca00 : 88 02 4c 66 fe ea ea ea  32
ca08 : ea ea a6 01 8e b5 03 a2  19
ca10 : 30 86 01 60 ae b5 03 86  81
ca18 : 01 60 20 9b bc a5 64 f0  31
ca20 : 03 4c a2 ca c6 65 a5 65  44
ca28 : c9 ff f0 1b c9 02 f0 22  45
ca30 : c9 03 90 07 c9 09 90 29  f9
ca38 : 4c 76 ca 0a aa bd 58 03  b3
ca40 : a8 bd 59 03 4c 95 b3 ac  17
ca48 : ad 03 a9 00 8d ad 03 4c  cc
ca50 : 95 b3 ad 0c c5 18 69 13  95
ca58 : a8 ad 0d c5 69 00 4c 98  cc
ca60 : ca 38 e9 03 aa bd 00 dc  73
ca68 : 29 10 d0 05 a0 01 4c a2  f7
ca70 : b3 a0 00 4c a2 b3 38 e9  79
ca78 : 09 0a aa bd 92 ca 85 fb  76
ca80 : bd 93 ca 85 fc a0 01 b1  a6
ca88 : fb 88 aa b1 fb a8 8a 4c  70
ca90 : 98 ca fa cb fc cb fe cb  87
ca98 : 84 63 85 62 a2 90 38 4c  a4
caa0 : 49 bc 85 66 a0 00 78 20  a2
caa8 : 0a ca b1 65 20 14 ca 58  af
cab0 : a8 4c a2 b3 20 ce cb 20  85
cab8 : 79 00 8d f8 cb f0 07 20  54
cac0 : 38 c9 85 fb 86 fc a0 00  90
cac8 : 20 db cb b0 59 20 b1 cb  d4
cad0 : d0 54 a2 6f 20 c6 ff ad  f4
```

```
cad8 : f8 cb d0 28 20 cf ff 85  7a
cae0 : fb 8d fa cb 20 cf ff 85  65
cae8 : fc 8d fb cb 20 cf ff 8d  bf
caf0 : fc cb 20 cf ff 8d fd cb  d0
caf8 : 20 cf ff 8d fe cb 20 cf  20
cb00 : ff 8d ff cb 20 cf ff 78  b1
cb08 : a0 00 20 0a ca 91 fb 20  5b
cb10 : 14 ca 58 20 b7 ff d0 0e  7e
cb18 : 20 a7 cb b0 09 e6 fb d0  6e
cb20 : e3 e6 fc 4c 04 cb a9 6f  63
cb28 : 20 c3 ff 20 cc ff 20 ac  d4
cb30 : c6 60 20 ce cb 20 38 c9  3a
cb38 : 85 fd 86 fe a0 01 20 db  87
cb40 : cb b0 e3 20 b1 cb d0 de  db
cb48 : ad fa cb ae fb cb 85 fb  67
cb50 : 86 fc a2 6f 20 c9 ff a5  86
cb58 : fb 20 d2 ff a5 fc 20 d2  80
cb60 : ff ad fc cb 20 d2 ff ad  e2
cb68 : fd cb 20 d2 ff ad fe cb  ae
cb70 : 20 d2 ff ad ff cb 20 d2  33
cb78 : ff 78 20 0a ca a0 00 b1  12
cb80 : fb 20 14 ca 58 20 d2 ff  bc
cb88 : 20 b7 ff d0 99 20 a7 cb  6f
cb90 : b0 94 a5 fc c5 fe d0 06  37
cb98 : a5 fb c5 fd f0 88 e6 fb  53
cba0 : d0 d7 e6 fc 4c 79 cb 38  e5
cba8 : ad 01 dc c9 7f f0 01 18  fa
cbb0 : 60 a9 08 85 ba 20 b4 ff  17
cbb8 : a9 6f 85 b9 20 96 ff 20  a8
cbc0 : a5 ff 8d f9 cb 20 ab ff  74
cbc8 : ad f9 cb c9 30 60 20 fb  1c
cbd0 : c5 20 9e ad 20 a3 b6 20  3d
cbd8 : bd ff 60 a9 6f a2 08 20  4f
cbe0 : ba ff 20 c0 ff 60 ff 60  7e
cbe8 : 49 54 45 53 a0 a0 a0 a0  ea
cbf0 : a0 a0 a0 a0 a0 00 00 00  27
cbf8 : 00 00 00 a0 80 00 05 00  29
```

**Listing 1. Hauptprogramm »Scroll-Machine« (Schluß)**

```
0 REM************* EDITOR **************    <154>
1 REM*-------------------------------*      <132>
2 REM*                               *      <229>
3 REM*   (C) 1985 BY THILO HERRMANN  *      <199>
4 REM*           POSTSTR. 6          *      <138>
5 REM*           7321 B£RTLINGEN     *      <182>
6 REM*                               *      <233>
7 REM*           TEL. 07161/52592    *      <050>
8 REM*-------------------------------*      <139>
9 REM*********************************      <044>
10 :                                        <068>
100 IF PEEK(50000)=56 AND PEEK(50001)=253
    THEN 160                                <086>
110 PRINT"SCROLL-MACHINE LADEN !"           <193>
120 END                                     <248>
130 :                                       <188>
140 :                                       <198>
150 REM ERWEITERUNG EIN                     <082>
160 SYS 49152                               <067>
170 REM BASIC-SPEICHER UEBER 20000          <128>
180 REM         FREIHALTEN                  <037>
190 POKE 56,78:POKE 55,32:CLR               <253>
200 ZE=160:V=53248:AS=1024:SP=53240:M=1     <054>
210 SX=40:SY=180:TT=197:NT=64               <241>
220 XK=64:YK=8                              <099>
230 H1=8:H2=16:H3=255:H4=256                <030>
240 UF=55296: REM ZWISCHENSPEICHER          <207>
250 A(0)=1:A(1)=0:A(2)=128                  <201>
260 POKE V+21,0                             <091>
270 A=3:GOSUB 2240                          <196>
280 GOSUB 2280                              <114>
290 :                                       <093>
300 REM *** EINGABE ***                     <163>
310 GOSUB 2500:←US0                         <252>
320 PRINT"{CLR,WHITE}"                      <182>
330 SA=40960:XD=128:YD=64                   <170>
340 GOSUB 4860                              <180>
350 :                                       <153>
360 :                                       <163>
370 REM ZEICHENSATZ KOPIEREN                <167>
380 PRINT"{CLR}":GOSUB 5010                 <122>
390 :                                       <193>
400 REM *** PARAMETER SETZEN ***            <150>
410 ←RS:←ZF14                               <061>
420 S2=SA-19-7*XD:GOSUB 2130                <060>
430 ←MTS2,XD,YD                             <197>
440 ←JS1,64,64                              <018>
450 ←GM1:←GRXD*8,XD*16-8,0,YD*8-8           <067>
460 DF=3*XD                                 <132>
470 :                                       <017>
480 REM TEXTFENSTER VERLEGEN                <036>
490 ←RZ40,50,164                            <178>
500 ←SU0:←1R31,23,16                        <208>
510 ←RF6,6,6:←VI52224,15                    <185>
520 :                                       <068>
530 IF WH THEN 760                          <130>
540 ←FU52224,53223,32                       <211>
550 ←US1:GOSUB 2500:GOSUB 3080              <128>
560 F(1)=14:F(4)=6                          <254>
570 IF C=1 THEN F(1)=0:F(2)=7:F(3)=8:F(4)=
    10                                      <048>
580 GOSUB 2500:GOSUB 2930                   <047>
590 :                                       <138>
600 REM PALETTE ERSTELLEN                   <166>
610 A=52867:B=40:C=32                       <137>
620 FOR I=0 TO 7                            <245>
630 FOR T=0 TO 31                           <055>
640 POKE A+I*B+T,I*C+T                      <155>
650 NEXT T,I                                <215>
660 :                                       <208>
670 REM *** LOESCHEN ***                    <124>
680 GOSUB 2500:PRINT"LOESCHEN (J/N)?"       <189>
690 GOSUB 2060:ON A GOTO 700,760            <254>
700 PRINT"{UP}WIRKLICH (J/N)?"              <224>
710 GOSUB 2060:ON A GOTO 720,760            <020>
720 ←FUSA,EA,32                             <163>
730 :                                       <022>
740 :                                       <032>
750 REM *** SPRITES ***                     <165>
760 ←FUAS,AS+62,.                           <180>
770 FOR T=AS+6 TO T+21 STEP 3:POKE T,3:NEX
    T                                       <033>
780 FOR T=AS+8 TO T+21 STEP 3:POKE T,192:N
    EXT                                     <150>
790 POKE AS+31,255:POKE AS+4,255            <037>
800 ←UT AS,AS+64,55296,48,48                <218>
810 POKE SP,96:POKE SP+1,96                 <236>
820 POKE V,175:POKE V+1,100                 <172>
830 POKE V+2,0:POKE V+3,180                 <047>
840 POKE V+40,7                             <169>
```

**Listing 2. Editor »Scroll-Machine«. Zur sicheren Eingabe verwenden Sie bitte den Checksummer.**

```
850 POKE V+21,3                                 <174>
860 :                                           <153>
870 ←FU960,1023,0                               <080>
880 ←FU1024,2023,32                             <184>
890 ←FU2040,2047,15                             <200>
900 GOSUB 2570                                  <226>
910 :                                           <203>
920 IF WH THEN 970                              <012>
930 WH=1:REM FLAG SETZEN                        <227>
940 :                                           <233>
950 REM AUF 0,0 -POSITION SETZEN                <043>
960 ←POXD*8,0                                   <157>
970 ←TM1:←US1:PRINT                             <067>
980 E$="_":GOSUB 4760:GOTO 1070                 <088>
990 :                                           <027>
1000 :                                          <037>
1010 REM ***** HAUPTROUTINE *****               <208>
1020 :                                          <057>
1030 REM *** TASTATURABFRAGE ***                <245>
1040 REM CURSORTASTEN ABFRAGEN                  <134>
1050 GET E$:IF E$=""THEN 1490                   <193>
1060 IF E$="{UP}"OR E$="{DOWN}"OR E$="{LEF
     T}"OR E$="{RIGHT}"THEN GOSUB 2730:GOT
     O 1410                                     <036>
1070 E=VAL(E$)                                  <168>
1080 REM ZEICHEN UNTER CURSOR LOESCHEN          <142>
1090 IF PEEK(TT)=.OR E$=CHR$(20)THEN POKE
     PO,32:GOTO 1490                            <176>
1100 IF E$=" "THEN GOSUB 4300                   <053>
1105 IF E$="@"THEN 5430                         <242>
1110 IF E$="R"THEN GOSUB 4370                   <152>
1120 IF E>0 THEN:←ASE-1                         <010>
1130 IF E$="{HOME}"THEN ZE=32                   <200>
1140 GOSUB 2500                                 <204>
1150 ←FU1024,2023,32                            <199>
1160 IF E$="L"THEN 680                          <007>
1170 :                                          <208>
1180 :                                          <218>
1190 GOSUB 2690                                 <008>
1200 IF E$="{F7}"THEN GOSUB 2250                <035>
1210 IF E$="K"THEN GOSUB 2200                   <235>
1220 IF E$="M"THEN GOSUB 4820                   <001>
1230 IF E$="M"THEN GOSUB 5230                   <135>
1240 IF E$="{F1}"THEN 1790                      <195>
1250 IF E$="{F3}"THEN 1890                      <207>
1260 IF E$="{CTRL-X}"THEN POKE 648,4:←US0:
     PRINT"{CLR}";:END                          <066>
1270 IF E$="F"THEN GOSUB 2910                   <042>
1280 IF E$="Z"THEN 3140                         <184>
1290 IF E$="X"THEN GOSUB 3420                   <205>
1300 IF E$="S"THEN GOSUB 3230                   <082>
1310 IF E$="C"THEN GOSUB 5010                   <074>
1320 IF E$="W"THEN GOSUB 4700                   <109>
1330 IF E$="A"THEN 5080                         <214>
1340 IF E$="U"THEN GOSUB 3570                   <131>
1350 IF E$="T"THEN 5540                         <254>
1360 IF E$="K"THEN GOSUB 3700                   <008>
1370 IF E$="↑"THEN GOSUB 5360                   <169>
1390 IF E$="←"THEN GOSUB 4760                   <193>
1400 REM BILDSCHIRM ZURUECKSETZEN               <239>
1410 GOSUB 2570:PRINT                           <181>
1420 ←FU55896,56295,FS                          <152>
1430 ←FU55296,55855,F(4)                        <207>
1440 POKE V+39,1:POKE 198,0                     <212>
1450 XC=ZE AND 31:YC=INT(ZE/32)                 <237>
1460 UT=1:GOSUB 2780:UT=0                       <162>
1470 PRINT TAB(26)"{UP,RIGHT,WHITE}NR. :"Z
     E"{LEFT,2SPACE}"                           <015>
1480 :                                          <007>
1490 GOSUB 1580                                 <050>
1500 IF M2 THEN GOSUB 1630                      <233>
1510 IF USR(4)THEN:GOSUB 1540:POKE PO,ZE        <037>
1520 ←SY1                                       <049>
1530 GOSUB 1690:GOTO 1050                       <229>
1540 IF M1 THEN ZE=K(PA AND 7,KM):PA=USR(K
     D)                                         <246>
1550 IF WE THEN ZE=ZE+1:IF ZE>W2 THEN ZE=W
     1                                          <203>
1560 RETURN                                     <172>
1565 :                                          <093>
1570 REM ADRESSE BERECHNEN                      <072>
1580 PO=USR(3)-DF                               <243>
1590 IF PO<.THEN PO=65536+PO                    <145>
1600 RETURN                                     <212>
1610 :                                          <138>
1620 REM MICRO-PUNKT BERECHNEN                  <172>
1630 PU=P2((P1 AND 1)+(P2 AND 1)*2)             <123>
1640 ZE=S(R(USR(PO))OR PU)                      <002>
1650 P1=USR(1)/4:P2=USR(2)/4                    <078>
1660 RETURN                                     <016>
1670 :                                          <198>
1680 REM X- UND Y-KOORDINATEN PRINTEN           <117>
1690 GOSUB 1750                                 <250>
1700 PRINT"{UP}X:"XP;                           <141>
1710 PRINT"{LEFT,SPACE}Y:"YP;                   <165>
1720 PRINT"{LEFT,SPACE}ADDR:"PO"{LEFT,3SPA
     CE}"                                       <013>
1730 RETURN                                     <086>
1740 :                                          <012>
1750 XP=INT(USR(1)/8-XD)                        <161>
1760 YP=INT(USR(2)/8):RETURN                    <046>
1770 :                                          <042>
1780 REM *** ABSPEICHERN ***                    <169>
1790 PRINT"{RIGHT}ABSPEICHERN (J/N)?"           <121>
1800 GOSUB 2060:ON A GOTO 1810,1410             <181>
1810 GOSUB 5780:IF A$="K"THEN 3960              <094>
1820 ←MTSA,XD,YD                                <072>
1830 ←US0                                       <100>
1840 OPEN 1,8,15,"S:"+N$:GOSUB 1990             <176>
1850 ←SV N$,EA                                  <109>
1860 GOSUB 1980:GOTO 420                        <003>
1870 :                                          <143>
1880 REM *** LADEN ***                          <078>
1890 PRINT"{RIGHT}LADEN (J/N)?"                 <030>
1900 GOSUB 2060:ON A GOTO 1910,1410             <026>
1910 GOSUB 5780:IF A$="K"THEN 4190              <190>
1920 ←LD N$                                     <232>
1930 GOSUB 1980:IF EN THEN 1950                 <190>
1940 SA=USR(10):XD=USR(11):YD=USR(12)           <048>
1950 GOTO 420                                   <196>
1960 :                                          <233>
1970 REM FLOPPYMELDUNG                          <046>
1980 ←US0:OPEN 1,8,15                           <250>
1990 INPUT#1,EN,ER$,ET,ES                       <177>
2000 CLOSE 1:←US1                               <026>
2010 IF ER$="OK"THEN RETURN                     <236>
2020 PRINT"{DOWN,RIGHT}"EN;ER$;ET;ES            <038>
2030 WAIT 197,63:POKE 198,0:RETURN              <084>
2040 :                                          <057>
2050 REM J/N EINGABE                            <075>
2060 POKE 198,0                                 <169>
2070 GET A$                                     <036>
2080 IF A$="J"THEN A=1:RETURN                   <203>
2090 IF A$<>"N"THEN 2070                        <136>
2100 A=2:RETURN                                 <041>
2110 :                                          <128>
2120 REM ENDADRESSE BERECHNEN                   <071>
2130 EA=SA+XD*YD-1:RETURN                       <089>
2140 :                                          <158>
2150 :                                          <168>
2160 REM HI/LO-BYTE UMRECHNUNG                  <191>
2170 H=INT(A/256):L=A-H*256:RETURN              <138>
2180 :                                          <198>
2190 REM KONTUR EINSTELLEN                      <251>
2200 GOSUB 3880                                 <000>
2210 INPUT"GESCHWINDIGKEIT:{2RIGHT}8{3LEFT
     }";GE                                      <022>
2220 IF A<3 THEN:XK=GE:YK=64:GOTO 2240          <173>
2230 YK=GE:XK=64                                <153>
2240 KM=A:KD=(A-1)/2+1                          <246>
2250 WE=0:M1=1:M2=0:←JS1,XK,YK:RETURN           <084>
2260 :                                          <022>
2270 REM MICRO-DATAS                            <137>
2280 DATA 32,108,123,98,124,225,255,254,12
     6,127,97,252,226,251,236,160               <019>
2290 DIM R(255),S(15)                           <245>
2300 RESTORE                                    <144>
2310 FOR I=0 TO 15                              <197>
2320 READ A:S(I)=A:R(A)=I:NEXT                  <208>
2330 :                                          <093>
2340 FOR I=0 TO 3:P2(I)=2↑(3-I):NEXT            <204>
2350 :                                          <113>
2360 REM Y - DATAS                              <195>
2370 DATA 100,111,121,98,248,247,227,160        <152>
2380 FOR I=7 TO 0 STEP-1:READ A:K(I,3)=A        <212>
2390 K(I,4)=A+128 AND 255:NEXT                  <227>
2400 :                                          <163>
2410 REM X - DATAS                              <244>
2420 DATA 101,116,117,97,246,234,231,160        <201>
2430 FOR I=0 TO 7:READ A:K(I,1)=A               <127>
```

```
2440 K(I,2)=A+128 AND 255:NEXT                  <019>
2450 :                                          <213>
2460 RETURN                                     <051>
2470 :                                          <233>
2480 :                                          <243>
2490 REM NORMAL SCHALTEN                        <109>
2500 POKE 648,4                                 <102>
2510 ←MC21,52,52                                <079>
2520 ←CI151,148,148                             <245>
2530 GOSUB 2650                                 <069>
2540 RETURN                                     <131>
2550 :                                          <057>
2560 REM PALETTE EINSCHALTEN                    <214>
2570 ←MC52,52,52                                <144>
2580 ←CI148,148,148                             <056>
2590 POKE 648,204                               <035>
2600 POKE 214,14:PRINT                          <196>
2610 ←FU55856,55895,6:REM HIN.FAR.              <190>
2620 RETURN                                     <212>
2630 :                                          <138>
2640 REM CURSOR SETZEN + ZEILE LOESCHEN         <149>
2650 POKE 214,14:PRINT:PRINT"{39SPACE,UP}"      <158>
2660 RETURN                                     <252>
2670 :                                          <178>
2680 REM AUF VIELFACHES VON 8 SETZEN            <203>
2690 GOSUB 1750:←POXD*8+XP*8,YP*8:RETURN        <085>
2700 :                                          <208>
2710 :                                          <218>
2720 REM WAHL DES CHARACTERS                    <033>
2730 IF E$="{UP}"THEN YC=YC-1:IF YC<0 THEN
     YC=7                                       <012>
2740 IF E$="{DOWN}"THEN YC=YC+1:IF YC>7 TH
     EN YC=0                                    <147>
2750 IF E$="{LEFT}"THEN XC=XC-1:IF XC<0 TH
     EN XC=31                                   <085>
2760 IF E$="{RIGHT}"THEN XC=XC+1:IF XC>31
     THEN XC=0                                  <220>
2770 REM SPRITESTEUERUNG                        <004>
2780 CX=XC*H1+SX                                <072>
2790 POKE V+2,CX AND H3                         <170>
2800 POKE V+H2,SGN(CX AND H4)*2                 <224>
2810 POKE V+3,SY+YC*H1                          <017>
2820 :                                          <073>
2830 IF UT THEN 2860                            <196>
2840 IF PEEK(TT)<>NT THEN GET E$:GOTO 2730      <175>
2850 GOSUB 2570:PRINT                           <091>
2860 ZE=YC*32+XC:REM ZEICHEN BERECHNEN          <211>
2870 RETURN                                     <207>
2880 :                                          <133>
2890 :                                          <143>
2900 REM FARBENEINGABE                          <135>
2910 GOSUB 3080                                 <193>
2920 :                                          <173>
2930 PRINT                                      <022>
2940 A=1:GOSUB 3050:                            <116>
2950 ON C GOTO 2960,2980                        <190>
2960 A=2:GOSUB 3050:POKE V+34,F(2)              <124>
2970 A=3:GOSUB 3050:POKE V+35,F(3)              <137>
2980 ←2R200+B,200,192+B                         <130>
2990 A=4:GOSUB 3050:←HF6,6,F(1)                 <034>
3000 :                                          <253>
3010 FS=F(4)                                    <227>
3020 IF B=0 THEN FS=F(1)                        <064>
3030 GOSUB 2570:RETURN                          <004>
3040 :                                          <037>
3050 PRINT"{UP}FARBE"A":{2SPACE}"F(A)"{LEF
     T,SPACE}"                                  <059>
3060 PRINT"{UP}"TAB(10);:INPUT F(A):RETURN      <052>
3070 :                                          <067>
3080 PRINT"MULTICOLOR (J/N)?"                   <082>
3090 GOSUB 2060:GOSUB 2500:B=(1-(A-1))*16:
     C=A                                        <077>
3100 RETURN                                     <182>
3110 :                                          <108>
3120 :                                          <118>
3130 REM ZEICHENSATZ LADEN                      <129>
3140 PRINT"{2RIGHT}ZEICHENSATZ LADEN (J/N)
     ?"                                         <106>
3150 GOSUB 2060:ON A GOTO 3160,3190             <007>
3160 INPUT"{DOWN,RIGHT}NAME: ";N$               <099>
3170 REM LADEN AN 53248 - 2                     <085>
3180 ←LD N$,53246:GOSUB 1980                    <162>
3190 GOTO 420                                   <161>
```

**Listing 2. Editor »Scroll-Machine« (Fortsetzung)**

```
3200 :                                          <198>
3210 :                                          <208>
3220 REM SCHREIBEN                              <194>
3230 ←SY0:←TM0:←JS0:RT=USR(1)                   <128>
3240 X=USR(1):Y=USR(2)                          <118>
3250 GET A$:IF A$="{F7}"THEN:←TM1:←JS1:RET
     URN                                        <173>
3260 IF A$=""THEN 3250                          <031>
3270 A=ASC(A$)                                  <065>
3280 REM CURSORSTEUERCODES BEACHTEN             <201>
3290 IF A=32 THEN 3360                          <060>
3300 IF A=20 OR A=157 THEN X=X-8:GOTO 3380      <141>
3310 IF A=29 THEN X=X+8:GOTO 3380               <095>
3320 IF A=17 THEN Y=Y+8:GOTO 3380               <104>
3330 IF A=145 THEN Y=Y-8:GOTO 3380              <166>
3340 IF A=13 THEN Y=Y+8:X=RT:GOTO 3380          <099>
3350 IF A=18 OR A=146 THEN PRINT A$;:GOTO
     3250                                       <181>
3360 PRINT"{HOME}"A$;:A=PEEK(1024)              <164>
3370 POKE USR(3)-DF,A:X=X+8                     <041>
3380 ←POX,Y:GOTO 3240                           <168>
3390 :                                          <133>
3400 :                                          <143>
3410 REM WECHSEL                                <249>
3420 GOSUB 2570:GOSUB 2650                      <088>
3430 PRINT"WECHSEL VON":GOSUB 3500              <221>
3440 W1=ZE                                      <086>
3450 PRINT"{UP}WECHSEL BIS":GOSUB 3500          <109>
3460 W2=ZE:IF W2<W1 THEN 3450                   <103>
3470 WE=1:RETURN                                <226>
3480 :                                          <223>
3490 REM WAEHLEN DES ZEICHENS                   <119>
3500 GOSUB 2730                                 <018>
3510 GET E$:IF E$=CHR$(13)THEN RETURN           <094>
3520 IF E$=""THEN 3510                          <039>
3530 GOTO 3500                                  <040>
3540 :                                          <027>
3550 :                                          <037>
3560 REM UEBERTRAGEN                            <184>
3570 GOSUB 2570:GOSUB 2650                      <238>
3580 Y2=YC:X2=XC:UT=1                           <185>
3590 PRINT"UEBERTRAGEN !"                       <070>
3600 GET E$:IF E$=CHR$(13)THEN UT=.:RETURN      <124>
3610 IF E$=""THEN 3600                          <130>
3620 GOSUB 2730                                 <139>
3630 A=USR(3)-DF+XC-X2+YC*XD-Y2*XD              <037>
3640 IF A>=SA AND A<=EA THEN POKE A,ZE          <046>
3650 ←SY1                                       <140>
3660 GOTO 3600                                  <172>
3670 :                                          <158>
3680 :                                          <168>
3690 REM KONTUREN DEFINIEREN                    <086>
3700 GOSUB 3880:A2=A                            <130>
3710 GOSUB 2570:GOSUB 2650                      <123>
3720 PRINT"ERSTES ELEMENT ?"                    <146>
3730 GOSUB 3500                                 <245>
3740 GOSUB 2650                                 <004>
3750 :                                          <238>
3760 REM AUSTEIGEND ODER FALLEND                <086>
3770 PRINT"1...[-1] / 2...[+1]"                 <117>
3780 GET A$:A=VAL(A$)*2-3                       <060>
3790 IF A<-1 OR A>1 THEN 3780                   <131>
3800 A3=0                                       <060>
3810 REM IN VARIABLEN SCHREIBEN                 <061>
3820 FOR I=-3.5*A TO 3.5*A STEP A               <181>
3830 K(I+4,A2)=ZE+A3:A3=A3+1:NEXT               <064>
3840 A=A2:GOSUB 2650:GOTO 2210                  <145>
3850 :                                          <083>
3860 :                                          <093>
3870 REM WAHL DES KONTURMODES                   <031>
3880 PRINT"1...X+ B 2...X- B 3...Y+ B 4...
     Y-                                         <060>
3890 GET A$:A=VAL(A$)                           <239>
3900 IF A<1 OR A>4 THEN 3890                    <076>
3910 RETURN                                     <227>
3920 :                                          <153>
3930 :                                          <163>
3940 REM ABSPEICHERN AUF KASSETTE               <102>
3950 REM SCHREIBEN VON SA,XD,YD                 <183>
3960 ←OF:OPEN 1,1,1,N$:PRINT#1,SA","XD","Y
     D:CLOSE 1                                  <196>
3970 REM SCHREIBEN DES SCHIRMS                  <168>
3980 GOSUB 4050:A=SA:GOSUB 2170                 <067>
3990 POKE 251,L:POKE 252,H                      <042>
4000 A=EA+1:GOSUB 2170                          <148>
```

```
4010 POKE 780,251:POKE 781,L:POKE 782,H     <130>
4020 SYS 65496:CLOSE 1:SYS 49152:GOTO 410   <175>
4030 :                                      <007>
4040 REM OEFFNEN DES KASSETTENKANALS        <175>
4050 L=LEN(N$):POKE 183,L                   <074>
4060 S=256*204                              <207>
4070 IF L=0 THEN 4100                       <026>
4080 FOR X=1 TO L                           <171>
4090 POKE S+X-1,ASC(MID$(N$,X,1)):NEXT      <062>
4100 POKE 780,1:POKE 781,1:POKE 782,0       <067>
4110 SYS 65466                              <199>
4120 POKE 780,L:POKE 781,0                  <163>
4130 POKE 782,204:SYS 65469                 <018>
4140 RETURN                                 <202>
4150 :                                      <128>
4160 :                                      <138>
4170 REM LADEN VON KASSETTE                 <164>
4180 REM LESEN VON SA,XD,YD                 <130>
4190 ←OF:OPEN 1,1,0,N$:INPUT#1,S2,X2,Y2:CL
     OSE 1                                  <219>
4200 REM LESEN DES SCHIRMS                  <115>
4210 GOSUB 4050:A=S2:GOSUB 2170             <027>
4220 POKE 780,0:POKE 781,L:POKE 782,H       <237>
4230 SYS 65493                              <063>
4240 IF(ST AND 48)THEN PRINT"{DOWN}BAND FE
     HLER !":CLOSE 1:WAIT 197,63:GOTO 410   <036>
4250 SA=S2:XD=X2:YD=Y2:SYS 49152:GOTO 410   <208>
4260 :                                      <238>
4270 :                                      <248>
4280 REM SCHNELLGANG                        <125>
4290 REM SYNCRONISATION AUS + HF=GRUEN      <004>
4300 ←SY0:←RF6,6,5:POKE V+39,1              <219>
4310 GOSUB 1580:GOSUB 1690                  <216>
4320 GET E$:IF E$=""THEN 4310               <141>
4330 E$="":←RF6,6,6:RETURN                  <076>
4340 :                                      <062>
4350 :                                      <072>
4360 REM RECHTECK AUSFUELLEN                <229>
4370 GOSUB 4480:REM BEREICH WAEHLEN         <063>
4380 IF E$="←"THEN RETURN                   <171>
4390 :                                      <113>
4400 REM RECHTECK FUELLEN                   <036>
4410 FOR I=YS TO YP                         <192>
4420 BR=PS+(YP-I)*XD                        <030>
4430 ←FU BR,BR+(XP-XS),ZE                   <113>
4440 NEXT                                   <235>
4450 E$="":RETURN                           <154>
4460 :                                      <183>
4470 REM BEREICH WAEHLEN                    <012>
4480 XS=XP:YS=YP:PS=PO:G=USR(PS)            <227>
4490 POKE PS,(USR(PS)+128)AND H3            <108>
4500 POKE V+39,PEEK(V+39)+1 AND H3          <088>
4510 ←SY1                                   <235>
4520 GOSUB 1580:GOSUB 1690                  <171>
4530 IF USR(4)THEN 4590                     <003>
4540 GET E$                                 <215>
4550 IF E$=""THEN 4490                      <057>
4560 IF E$=" "THEN GOSUB 4300               <198>
4570 IF E$="←"THEN POKE PS,G:GOTO 4760      <028>
4580 GOTO 4490                              <079>
4590 POKE PS,G                              <172>
4600 :                                      <067>
4610 REM NORMALISIEREN                      <123>
4620 IF XP-XS<0 THEN A=XP:XP=XS:XS=A        <227>
4630 IF YP-YS<0 THEN A=YP:YP=YS:YS=A        <243>
4640 PS=SA+YS*XD+XS                         <238>
4650 RETURN                                 <202>
4660 :                                      <128>
4670 :                                      <138>
4680 :                                      <148>
4690 REM UMSCHALTEN AUF WECHSELMODUS        <086>
4700 WE=WE+1 AND 1                          <019>
4710 IF WE THEN ZE=W1:RETURN                <231>
4720 GOTO 2860                              <219>
4730 :                                      <198>
4740 :                                      <208>
4750 REM RESET                              <178>
4760 M1=0:M2=0:GOSUB 2690:←JS1,64,64        <208>
4770 WE=0:ZE=160                            <084>
4780 RETURN                                 <076>
4790 :                                      <002>
4800 :                                      <012>
4810 REM MICRO - MODE SETZEN                <016>
4820 M2=1:M1=0:WE=0:←JS1,32,32:RETURN       <235>
4830 :                                      <042>
4840 :                                      <052>
4850 REM EINGABE VON AA,XD,YD               <132>
4860 GOSUB 2650                             <104>
4870 PRINT"{2RIGHT}X- UND Y-AUSDEHNUNG{3SP
     ACE}"XD","YD                           <189>
4880 PRINT"{UP}"SPC(23);:INPUT XD,YD        <036>
4890 PRINT"{DOWN,2RIGHT}STARTADRESSE{3SPAC
     E}"SA                                  <126>
4900 PRINT"{UP}"SPC(16);:INPUT SA           <105>
4910 IF XD<1 OR YD<1 THEN 4860              <246>
4920 GOSUB 2130                             <158>
4930 PRINT"{DOWN,2RIGHT}START:"SA"{2SPACE}
     ENDE:"EA                               <249>
4940 IF SA<20000 OR(SA>49152 AND SA<57344)
     OR SA>65530 THEN 4860                  <220>
4950 IF(EA>49152 AND EA<57344)OR EA>65535
     THEN 4860                              <216>
4960 PRINT"{DOWN,3RIGHT}RICHTIG (J/N)?"     <025>
4970 GOSUB 2060:ON A GOTO 4980,4860         <058>
4980 RETURN                                 <021>
4990 :                                      <203>
5000 REM ZEICHENSATZ VOM ROM INS RAM        <028>
5010 GOSUB 2500:PRINT"{2RIGHT}ZEICHENSATZ
     KOPIEREN (J/N)?"                       <070>
5020 GOSUB 2060:ON A GOTO 5030,5040         <086>
5030 ←CP                                    <171>
5040 RETURN                                 <081>
5050 :                                      <007>
5060 :                                      <017>
5070 REM UMFORMATIEREN EINES SCHIRMS        <219>
5080 SF=SA:EF=EA:XF=XD:YF=YD                <021>
5090 GOSUB 4860:←US0:L=48                   <105>
5100 L1=0:L2=YF-1:L3=1                      <223>
5110 IF XD>XF THEN L1=YF-1:L2=0:L3=-1       <177>
5120 :                                      <078>
5122 PRINT"{CLR,SPACE}SCHIRM AN NEUE ADRES
     SE KOPIEREN (J/N)?"                    <195>
5123 GOSUB 2060:ON A GOTO 5140,5190         <198>
5130 :                                      <088>
5140 FOR I= L1 TO L2 STEP L3                <107>
5150 A=SF+XF*I:B=SF+XF*(I+1)-1:C=B-A        <100>
5160 ←UT A,B,UF,L,L                         <170>
5170 ←UT UF,UF+C,SA+XD*I,L,L                <136>
5180 NEXT                                   <210>
5190 GOTO 420                               <121>
5200 :                                      <158>
5210 :                                      <168>
5220 REM MICRO-ZEICHEN DEFINIEREN           <141>
5230 GOSUB 2570:GOSUB 2650                  <113>
5240 PRINT"ERSTES ELEMENT ?"                <136>
5250 GOSUB 3500:REM ZEICHEN WAEHLEN         <190>
5260 IF ZE>240 THEN 5250                    <132>
5270 GOSUB 5320                             <001>
5280 FOR I=0 TO 15:S(I)=I+ZE:R(I+ZE)=I      <063>
5290 NEXT                                   <064>
5300 GOTO 4820                              <031>
5310 :                                      <012>
5320 FOR I=0 TO 255:R(I)=.:NEXT:RETURN      <082>
5330 :                                      <032>
5340 :                                      <042>
5350 REM KONTUR UND MICRO RESET             <080>
5360 PRINT"KONTUR UND MICRO ZURUECKSETZEN
     (J/N)?"                                <126>
5370 GOSUB 2060:ON A GOTO 5380,5390         <197>
5380 GOSUB 5320:GOSUB 2300                  <252>
5390 RETURN                                 <177>
5400 :                                      <103>
5410 :                                      <113>
5420 REM MICRO-MODE UMSCHALTEN              <144>
5430 R(32)=15-R(32)                         <030>
5440 FOR I=0 TO 15                          <011>
5450 R(S(I))=15-I                           <251>
5460 NEXT                                   <235>
5470 FOR I=0 TO 7                           <250>
5480 A=S(I):S(I)=S(15-I):S(15-I)=A          <095>
5490 NEXT                                   <009>
5500 GOTO 1490                              <232>
5510 :                                      <213>
5520 :                                      <223>
5530 REM TRANSFER EINES BEREICHES           <161>
5540 GOSUB 4480:REM BEREICH WAEHLEN         <213>
5550 IF E$="←"THEN 420                      <073>
5560 GOSUB 2650:PRINT"WOHIN (LINKER OBERER
     PUNKT) ?{UP}"                          <250>
5570 IF USR(4)THEN 5570                     <022>
```

```
5580 PT=PS:XT=XS:YT=YS                             <098>
5590 XR=XP:YR=YP:GOSUB 4480                        <198>
5600 IF E$="←"THEN 420                             <123>
5610 :                                             <057>
5620 REM RECHTECK UEBERTRAGEN                      <021>
5630 ←US0                                          <074>
5640 IF PO>PT THEN L1=YT:L2=YR:L3=1:GOTO 5
     660                                           <101>
5650 L1=YR:L2=YT:L3=-1                             <096>
5660 L=48                                          <156>
5670 FOR I=L1 TO L2 STEP L3                        <127>
5680 BR=(YR-I)*XD                                  <196>
5690 A=(XR-XT)                                     <149>
5700 ←UT PT+BR,PT+BR+A,UF,L,L                      <244>
5710 IF PO+BR+A>EA THEN 5730                       <100>
5720 ←UT UF,UF+A,PO+BR,L,L                         <188>
5730 NEXT                                          <250>
5740 GOTO 420                                      <161>
5750 :                                             <198>
5760 :                                             <208>
5770 REM EINGABE FUER DISK & KASSETTE              <001>
5780 N$="":INPUT"{DOWN,2RIGHT}NAME: ";N$           <104>
5790 PRINT"{DOWN,RIGHT,RVSON}D{RVOFF}ISK O
     DER{SPACE,RVSON}K{RVOFF}ASSETTE ?             <221>
5800 GET A$:IF A$<>"K"AND A$<>"D"THEN 5800         <085>
5810 RETURN                                        <086>
5820 GOSUB 2280:RESTORE                            <241>
5830 FOR I=0 TO 15                                 <146>
5840 R(S(I))=15-I:READ A:S(15-I)=A+128 AND
      255                                          <063>
5850 NEXT                                          <114>
5860 FOR I=0 TO 15:PRINT S(I);:NEXT                <103>
```

**Listing 2. Editor »Scroll-Machine« (Schluß)**

```
90 PRINT"{CLR,GREY 1}"                             <097>
100 SYS 49152                                      <007>
105 PRINT                                          <002>
110 AA=40960:XD=128:YD=64                          <187>
120 SP=32:ZE=207:FA=0                              <249>
130 SR=57344:REM SPRITEADRESSE                     <135>
140 V=53248:REM VIC                                <069>
145 CO=V+31:REM COLLISION                          <174>
150 SZ=53240:REM SPRITE-ZEIGER                     <134>
160 G1=AA+2470:G2=AA+2477                          <040>
165 G3=AA+6250:G4=AA+6271                          <045>
170 FZ=3:REM FASSZAEHLER                           <048>
180 A$(0)="AUF":A$(1)="{SPACE,WHITE,RVSON}
    ZU{GREY 1}"                                    <224>
190 DEF FN A(X)=SIN(X*.7)*COS(X*2.2)               <095>
195 :                                              <253>
200 ←RS   :REM RESET ALLER REGISTER                <237>
201 ←CP   :REM ZEICHENSATZ KOPIEREN                <041>
205 ←MT AA,XD,YD :REM                              <169>
206 :                                              <008>
300 REM FARBEN+RASTERZEILEN                        <045>
305 ←HF 0,0,11                                     <057>
310 ←RF 0,0,0                                      <022>
315 ←RZ 81,89,248                                  <023>
320 :                                              <123>
321 ←JS 2 :REM JOYSTICKSTEUERUNG                   <107>
322 ←TG 2000,2000                                  <237>
330 :                                              <133>
332 IF EG THEN 950                                 <169>
333 :                                              <136>
335 ←US0                                           <135>
340 INPUT"{2DOWN,2RIGHT}BESCHLEUNIGUNG (1-
```

**Listing 3. Ein kleines Spiel als Demoprogramm**



```
    255):{2RIGHT}16{4LEFT}";G                  <169>
343 INPUT"{2DOWN,2RIGHT}REIBUNG (1-255):{2
    RIGHT}8{3LEFT}";R                          <228>
347 INPUT"{2DOWN,2RIGHT}ENERGIE:{2RIGHT}50
    00{6LEFT}";EG                              <046>
350 PRINT"{2DOWN,2RIGHT}NEUER STOLLEN (J/N
    )?"                                        <143>
355 GET A$:IF A$="N"THEN 950                   <029>
360 IF A$<>"J"THEN 355                         <143>
365 INPUT"{2DOWN,2RIGHT}BREITE (3-20):{2RI
    GHT}7{3LEFT}";B                            <096>
480 :                                          <027>
485 PRINT TAB(10)"{2DOWN}*** STOLLENAUFBAU
     ***                                       <193>
490 ←FU AA,AA+8191,160                         <085>
500 H=XD/12                                    <155>
510 FOR I=30 TO XD-1-B                         <231>
515 P=INT(FN A(I/H)*16+32)*XD+AA+I             <011>
520 ←FUP,P+B,32                                <077>
525 ←FUP-XD,P+B-XD,SP                          <030>
526 ←FUP+XD,P+B+XD,SP                          <029>
530 NEXT I                                     <223>
590 :                                          <138>
600 FOR I=13 TO 47                             <073>
610 P=AA+30+I*XD                               <048>
615 IF I<20 THEN:←FU P,P+25,SP                 <217>
620 ←FU P,P+6,SP                               <227>
630 NEXT I                                     <067>
690 :                                          <238>
700 FOR I=19 TO 48                             <180>
710 P=AA+115+I*XD                              <200>
715 IF I>40 THEN:←FU P-10,P+13,SP              <070>
720 ←FU P,P+13,SP                              <117>
730 NEXT I                                     <167>
790 :                                          <083>
950 ←ZF0                                       <232>
955 ←US1                                       <246>
956 EE=EG                                      <135>
957 ←TBG,G:←TRR,R                              <126>
960 :                                          <253>
990 REM SPRITES                                <154>
1000 POKE SZ,128                               <246>
1010 POKE SZ+1,129                             <220>
1020 POKE V+39,15                              <148>
1030 POKE V+40,0                               <097>
1040 POKE V,172:POKE V+1,150                   <139>
1050 POKE V+2,171:POKE V+3,161                 <116>
1090 :                                         <128>
1100 POKE V+21,3                               <169>
1102 ←FU SR,SR+128,0                           <099>
1105 :                                         <143>
1107 REM SPRITE EINLESEN                       <016>
1108 RESTORE                                   <228>
1110 FOR I=25 TO I+18                          <061>
1120 READ A:POKE SR+I,A:NEXT                   <190>
1125 :                                         <163>
1127 REM ZEICHEN 81 IN SPRITE KOPIEREN         <052>
1130 FOR I=0 TO 7                              <245>
1140 POKE SR+65+I*3,USR(V+ZE*8+I)              <069>
1150 NEXT                                      <004>
1190 :                                         <228>
1195 REM FAESSER SETZEN                        <032>
1200 FOR I=G1+3 TO I+11 STEP 4                 <091>
1210 POKE I,ZE:NEXT                            <037>
1250 FOR I=G3 TO G4                            <251>
1260 POKE I,SP:NEXT                            <091>
1270 :                                         <052>
1300 ←PO 822,247                               <124>
1500 POKE V+21,1                               <056>
1900 PRINT"{HOME}FAESSER:"FZ" ENERGIE:"EE"
     {LEFT,SPACE}GREIFER: "A$(FA)              <169>
1902 IF USR(4)THEN 1902                        <175>
1905 ←JS2                                      <166>
1910 ←TM0                                      <173>
1915 GOSUB 6000                                <213>
1920 POKE CO,0                                 <012>
1930 IF FZ=0 THEN 6500                         <016>
1990 :                                         <007>
1995 REM **** HAUPTROUTINE ****                <089>
1996 :                                         <013>
2000 EE=EE-1:PRINT"{HOME}"TAB(20)EE"{LEFT,
     SPACE}"                                   <060>
2010 IF PEEK(CO)THEN 3000                      <123>
2020 IF USR(4)THEN 4000                        <029>
2500 IF EE>0 THEN 2000                         <044>
2590 :                                         <098>
2595 :                                         <103>
2600 PRINT"{CLR,2DOWN,2RIGHT}BATTERIE LEER
      => ABSTURZ                               <250>
2610 GOTO 3010                                 <137>
2990 :                                         <243>
3000 ←JS0:GOSUB 6000                           <124>
3002 IF PEEK(CO)=2 THEN 3100                   <068>
3005 PRINT"{CLR,2DOWN,RIGHT}SIE HABEN IHRE
     N TRANSPORTER ZERSTOERT.                  <110>
3010 ON FA+1 GOTO 7050,7020                    <010>
3100 PRINT"{CLR,2DOWN,5RIGHT}SIE HABEN DAS
      FASS ZERSTOERT.                          <109>
3110 GOTO 7020                                 <132>
3900 :                                         <133>
4000 EE=EE-10                                  <129>
4010 ON FA GOTO 4200                           <032>
4112 :                                         <090>
4115 P=USR(3):REM FASS AUFNEHMEN               <237>
4120 IF USR(P)<>ZE THEN 1910                   <128>
4122 ←JS0:REM TRAEGHEIT LOESCHEN               <013>
4124 REM Y-POSITION BERICHTIGEN                <218>
4125 ←PO USR(1),USR(2)AND 8184                 <078>
4127 ←TM1:REM BILDSCHIRM ABTASTEN              <092>
4130 FA=1:POKE V+21,3:REM SPRITE AN            <094>
4140 POKE P,SP:REM CHARACTER LOESCHEN          <153>
4145 IF P>G3 AND P<G4 THEN FZ=FZ+1             <232>
4150 GOTO 1900                                 <153>
4190 :                                         <168>
4200 P=USR(3):REM FASS ABSETZEN                <007>
4210 IF USR(P+XD)<>160 THEN 4300               <020>
4215 ←JS0                                      <179>
4220 ←PO(USR(1)AND 8184)+4,USR(2)AND 8184      <096>
4230 ←TM1:FA=0:POKE P,ZE                       <086>
4240 POKE V+21,1                               <247>
4245 IF P>G3 AND P<G4 THEN FZ=FZ-1             <077>
4250 GOTO 1900                                 <253>
4290 :                                         <012>
4300 PRINT"{CLR,2DOWN,2RIGHT}SIE HABEN DAS
      FASS FALLENLASSEN"                       <100>
4350 GOTO 7020                                 <096>
4900 :                                         <113>
4980 :                                         <193>
4990 REM SPRITEDATEN                           <099>
5000 DATA 255,0,3,255,192,60,231,60,239,25
     5,247,3,0,192,1,0,128,0,129               <109>
6000 FOR I=1 TO 20:NEXT:RETURN                 <190>
6500 PRINT"{CLR,2DOWN,3RIGHT}SIE HABEN IHR
     E MISSION ERFUELLT.                       <094>
6510 PRINT"{2DOWN,2RIGHT}RESTENERGIE:"EE       <193>
6520 PRINT"{2DOWN,2RIGHT}DAFUER BEKOMMEN S
     IE EIN PAAR ORDEN !                       <039>
6590 GOTO 7090                                 <048>
7020 PRINT"{3DOWN,4RIGHT}DER STOLLEN IST N
     UN VOLLSTAENDIG                           <082>
7030 PRINT"{DOWN,2RIGHT}STRAHLENVERSEUCHT.     <223>
7040 PRINT"{DOWN,2RIGHT}EIN FORTFUEHREN DE
     R ARBEIT WIRD IN                          <036>
7045 PRINT"{DOWN,2RIGHT}CA. 1500 JAHREN WI
     EDER MOEGLICH SEIN.                       <059>
7050 PRINT"{2DOWN,4RIGHT}S I E{3SPACE}S I
     N D{3SPACE}T O T{3SPACE}! ! !             <095>
7090 PRINT TAB(15)"{4DOWN}E{2SPACE}N{2SPAC
     E}D{2SPACE}E                              <026>
7100 ←US0                                      <014>
7101 POKE V+21,0                               <046>
7105 GET A$:IF A$=""THEN 7105                  <118>
7110 GOTO 110                                  <252>
```



**Listing 3. Ein kleines Spiel als Demoprogramm (Schluß)**

# Kurvenplotten mit Hardcopy

**Mit Hilfe dieses Programmes ist es auf dem C 16 möglich, eine oder mehrere Funktionen in ein Koordinatensystem zu plotten und anschließend eine Hardcopy herzustellen.**

Da der C 16 bei hochauflösender Grafik leider nur noch 2 KByte für Basic übrig hat, mußte auf REM-Zeilen verzichtet werden, wodurch die Übersichtlichkeit des Programms etwas leidet.

Das Programm gliedert sich in folgende Teile:

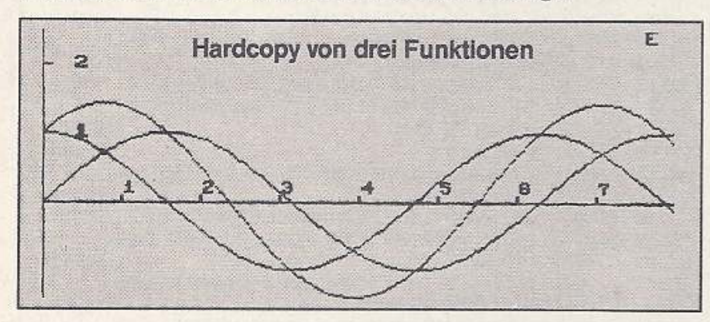
- 10-190 Initialisierung, Eingabe der Funktion, Wahl von Koordinatensystem und Maßstab
- 200-320 Funktion wird geplottet
- 330-390 Festlegung des Nullpunktes
- 400-480 Festlegung der Vergrößerung
- 510-610 Koordinatensystem und Beschriftung zeichnen
- 620-710 Bildschirm lesen und Ausgabe an Drucker

Nach dem Starten des Programmes mit RUN muß zunächst die gewünschte Funktion eingegeben werden. Dies geschieht dadurch, daß in der aufgelisteten Zeile 200 die gewünschte Funktion anstelle von F(x) eingegeben wird (x muß als Variable benutzt werden).

Durch Drücken von RETURN wird die Zeile programmiert. Wenn Sie wünschen, daß der hochauflösende Bildschirm gelöscht werden soll, so drücken Sie nun F1; soll die neue Funktion dagegen zu bestehenden Kurven hinzugeplottet werden, so drücken Sie F3.

Nun müssen Sie den Maßstab eingeben (für eine Sinusfunktion ist beispielsweise 40 ein guter Wert) und eines von den drei möglichen Koordinatensystemen auswählen.

Bei der Eingabe der Funktion braucht man nicht auf Definitionslücken zu achten, da die TRAP-Anweisung in Zeile 240 »DIVISION BY ZERO« Fehler und andere abfängt.



Hardcopy von drei Funktionen

Nach Beendigung des Plottens erscheint in der rechten oberen Ecke ein »E«. Drückt man jetzt auf »C«, wird der Bildschirm gelesen und als Hardcopy über den Drucker ausgegeben. Dies dauert leider fast 30 Minuten, da jeder der 64000 Bildpunkte einzeln ausgelesen und in ein dem Drucker genehmes Bitmuster umgewandelt werden muß. Dieser Programmteil (ab Zeile 620) wurde für den Seikosha GP100VC geschrieben und muß für andere Drucker eventuell geändert werden. Jeder andere Tastendruck führt zu einem kurzen Menü.

Man kann durch Druck auf F1 Maßstab und Koordinatensystem neu wählen.

Durch Druck auf F2 kommt man wieder zur Anfangsroutine und kann eine neue Funktion eingeben.

(Christian Schoske/ev)

```
10 REM CHRISTIAN SCHOSKE
20 KEY1,"O=1:GOTO90"+CHR$(13)
30 KEY 2,"RUN"+CHR$(13)
40 KEY 3,"O=0:GOTO90"+CHR$(13)
50 GRAPHIC0
60 PRINT"GEBEN SIE FUER F(X) DIE FUNKTI
ON EIN.    ANSCHLIESSEND RETURN DRUECKE
N"
70 PRINT"F1 BILDSCHIRM LOESCHEN":PRINT
"F3 BILDSCHIRM NICHT LOESCHEN":PRINT
80 PRINT"200 DEF FNA(X)=F(X)" :END
90 REM
100 GRAPHIC0:PRINT:PRINT:INPUT"MASSTAB "
;SC
110 PRINT:PRINT"KOORDINATENSYSTEM "
120 PRINT:PRINTSPC(3)" |"SPC(8)" |"SPC(9)"
 |"SPC(17)"1   |"SPC(8)" |  2"SPC(6)" |  3"
130 PRINT"____|____"SPC(4)" |"SPC(9)" |____
"SPC(13)" |"SPC(8)" |"SPC(9)" | "
140 PRINTSPC(3)" |"SPC(8)" |"SPC(9)" |"SPC(
20)" |"SPC(8)" |__________  |"
150 GETKEYA$:A%=VAL(A$):IFA%<1ORA%>3THEN
150
160 COLOR1,1,1
170 GRAPHIC1,0:CHAR1,38,0," "
180 ONA%GOTO 330,350,370
190 REM
200 DEF FNA(X)=SIN(X)
210 Z=0
220 DO
230 X=(-AX+Z)/SC
240 TRAP270
250 Y=SC*(FNA(X))
260 DRAW1,Z,AY-Y
270 Z=Z+1
280 LOOPWHILEZ<320
290 CHAR1,38,0,"E"
300 GETKEYA$:IFA$="C"THEN620:ELSEGRAPHIC
0
310 PRINT"F1 NEUER MASSTAB":PRINT"F2 NE
UE FUNTION"
320 END
330 AY=100:AX=160:DRAW1,0,AYTO320,AY:DRA
W1,AX,0TOAX,200
340 GOTO380
350 AY=199:AX=0:DRAW1,AX,0TOAX,AY:DRAW1,
AX,AYTO320,AY
360 GOTO380
370 AY=100:AX=0:DRAW1,AX,0TOAX,200:DRAW1
,AX,AYTO320,AY
380 GOSUB400:IFO<>0THENGOSUB510
390 GOTO190
400 WE =320/SC
410 IFWE >=230THENM=50
420 IFWE >=200ANDWE <230THENM=25
430 IFWE >=120ANDWE <200THENM=20
440 IFWE >= 65ANDWE <120THENM=10
450 IFWE >= 26ANDWE <  65THENM=5
460 IFWE >= 13ANDWE <  26THENM=2
470 IFWE<13THENM=1:F=1
480 RETURN
510 YA=AY:    XA=AX:MC=0:IFM>1THENF=1
520 DO
530 FORD=1TOM
540 XA=XA+SC:YA=YA-SC:IF(XA/F)>312THENEX
IT
550 NEXTD:MC=MC+1
560 IF(XA/F/8)-1>39THEN580
570 DRAW1,(XA/F),AYTO(XA/F),(AY-5):CHAR1
,(XA/F/8-1),(AY/8)-1,STR$(MC*M)
580 IF(YA/F/8)<1THEN600
590 DRAW1,AX,(YA/F)TO(AX+5),(YA/F):CHAR1
,(AX/8+1),(YA/F/8),STR$(MC*M)
600 LOOP
610 RETURN
620 OPEN4,4:Z=0:CMD4
630 FORA=0TO27
640 FORB=0TO31
650 FORC=0TO10
660 FORD=0TO6:LOCATE(B*10+C),(A*7+D):IFR
DOT(2)=1THENZ=Z+2^D
670 NEXTD
680 Z=Z+128:A$=A$+CHR$(Z):Z=0:NEXTC
690 PRINTCHR$(8)A$;:A$="":NEXTB
700 PRINT" ":NEXTA
710 PRINT#4,CHR$(15):CLOSE4:GOTO300

READY.
```

**Listing »Funktionenplotter C 16«**

# HI-EDDI — Druckerroutinen

**Schon lange warten HI-EDDI-Besitzer auf eine Anpassung für ihren Drucker. Die in Ausgabe 1/85 abgedruckte Version ließ eine Superhardcopy nur für den Epson zu. Doch jetzt kommen auch Besitzer von Commodore-Drucker in den Genuß, HI-EDDI-Bilder ausgeben zu können.**

Welcher 64'er Leser kennt ihn nicht — den HI-EDDI, das fantastische Zeichen- und Malprogramm aus Ausgabe 1/85. Jeder der einen Epson oder dazu kompatiblen Drucker besaß, kam sogar in den Genuß der ausgezeichneten Hardcopy-Funktion. Jetzt folgen Anpassungen für Drucker anderer Hersteller, diesmal für die Commodore-Drucker MPS 801/802/803 und kompatible. Für andere Drucker werden noch Lösungen gesucht.

## Hi-EDDI für MPS 802 und kompatible

Ich begann nach Überwindung meiner Abneigung gegen stundenlanges Tastenklopfen HI-EDDI abzutippen. Nach den ersten Tests dieses fantastischen Zeichen- und Malprogramms, bereute ich es aber nicht mehr mich dieser Tortur unterzogen zu haben. Doch mußte ich als Besitzer eines MPS-802-Druckers feststellen, was ist HI-EDDI ohne Superhardcopy?

Die Antwort darauf veranlaßte mich dazu, die Anregung in der Funktionsbeschreibung aufzugreifen und eine Lösung für einen anderen Drucker zu suchen. Gesagt — getan, hier ist mein Lösungsvorschlag für den Commodore MPS-802 (Listing 1)

Die Druckerroutine hat dieselben Eigenschaften wie die in 1/85 veröffentlichte und von Hans Haberl geschriebene Routine für Epson-Drucker. Daher kann auf eine Funktionsbeschreibung verzichtet werden. Ich hoffe, daß der Ausdruck von HI-EDDI-Grafikbildern mit diesem Programm auf allen Versionen des Commodore 1526/MPS-802 Druckers funktioniert. Bei meiner Ausführung (REV 07C) konnte ich bis jetzt keine Fehler feststellen.

(Franz Illetschko/gk)

## HI-EDDI für MPS 801/803 und kompatible

Eine Lösung für den Commodore MPS 801/803 schickte uns Florian Kulzer (Listing 2). Auch hier braucht am HI-EDDI selbst nichts verändert zu werden. Die Hardcopy wird, wie im 64'er 1/85 beschrieben, mit den Tasten »Commodore« und »P« gestartet.

Allerdings ist mit diesem Programm das Nebeneinanderdrucken zweier Bilder nicht möglich. Ausgegeben wird immer eine Hardcopy des aktuellen Bildschirms. Daher können die Fragen nach Nummer und Format der Hardcopy beliebig beantwortet werden.

(Florian Kulzer/gk)

```
programm : hi-print          0d00 0f9b
--------------------------------------

0d00 : 85 5b a9 64 a2 04 a0 00  f7
0d08 : 20 08 0e a9 6a a2 04 a0  f2
0d10 : 06 20 08 0e a2 6a 20 c9  7c
0d18 : ff a9 14 20 d2 ff a2 64  75
0d20 : 20 c9 ff a9 8d 20 d2 ff  7f
0d28 : a9 69 a2 04 a0 05 20 08  72
0d30 : 0e a5 5b 0a 90 03 4c 86  88
0d38 : 0e a5 5b 29 07 a2 02 20  e3
0d40 : 13 0e a9 14 85 fb a5 5b  cd
0d48 : 29 c0 f0 10 a5 5b 4a 4a  02
0d50 : 4a 29 07 a2 00 20 13 0e  ae
0d58 : a9 00 85 fb a9 00 8d 3d  2d
0d60 : 03 20 2d 0e 20 66 0e f0  d0
0d68 : 29 a2 69 20 c9 ff a2 00  68
0d70 : bd 3e 03 20 d2 ff e8 e0  a4
0d78 : 08 d0 f5 a9 0d 20 d2 ff  b8
0d80 : a2 64 20 c9 ff 20 74 0e  84
0d88 : a9 fe 20 d2 ff a9 8d 20  d6
0d90 : d2 ff 18 a5 5e 69 08 85  79
0d98 : 5e 90 02 e6 5f 20 e1 ff  1a
0da0 : f0 3f ee 3d 03 ad 3d 03  2c
0da8 : c9 28 d0 b5 a5 5b 29 c0  cb
0db0 : f0 18 a5 5e a6 5c 85 5c  fe
0db8 : 86 5e a5 5f a6 5d 85 5d  e9
0dc0 : 86 5f a5 fb 49 28 85 fb  c2
0dc8 : d0 92 a2 64 20 c9 ff a9  ba
0dd0 : 0d 20 d2 ff ee 3c 03 ad  da
0dd8 : 3c 03 c9 19 f0 03 4c 5c  3c
0de0 : 0d a9 69 20 c3 ff a2 6a  bc
0de8 : 20 c9 ff a9 24 20 d2 ff  b0
0df0 : a2 64 20 c9 ff a9 8d 20  c9
0df8 : d2 ff a9 6a 20 c3 ff 20  e2
0e00 : cc ff a9 64 20 c3 ff 60  a4
0e08 : 20 ba ff a9 00 20 bd ff  b2
0e10 : 4c c0 ff a0 06 d9 96 1f  98
0e18 : f0 03 88 d0 f8 a9 00 95  ce
0e20 : 5c 8d 3c 03 18 69 20 88  11
0e28 : 10 fb 95 5d 60 78 a9 34  20
0e30 : 85 01 a2 00 8a 9d 3e 03  73
0e38 : e8 e0 08 d0 f8 a9 80 85  96
0e40 : 60 a0 00 b1 5e a2 00 0a  36
0e48 : 90 0a 48 bd 3e 03 05 60  78
0e50 : 9d 3e 03 68 e8 e0 08 d0  32
0e58 : ee 46 60 c8 c0 08 d0 e3  f2
0e60 : a9 37 85 01 58 60 a2 00  39
0e68 : 8a dd 3e 03 d0 05 e8 e0  6b
0e70 : 08 d0 f6 60 18 a5 fb 6d  23
0e78 : 3d 03 aa f0 08 a9 20 20  8e
0e80 : d2 ff ca d0 fa 60 a5 5b  1e
0e88 : 29 07 a2 00 20 13 0e 85  bb
0e90 : 5f a9 00 85 5e 85 fb 85  82
0e98 : 59 85 5a a9 c0 95 fc 4a  c1
0ea0 : 4a e8 e0 04 d0 f7 a9 00  8a
0ea8 : 8d 3d 03 20 2d 0e a5 5a  27
0eb0 : f0 11 a2 00 bd 3e 03 0a  bf
0eb8 : 0a 0a 0a 9d 3e 03 e8 e0  5f
0ec0 : 08 d0 f1 20 62 0f 4c d6  2e
0ec8 : 0e a2 00 bd 4a 03 9d 46  9f
0ed0 : 03 e8 e0 04 d0 f5 20 8d  58
0ed8 : 0f f0 2f a2 69 20 c9 ff  3e
0ee0 : a2 00 bd 46 03 20 d2 ff  37
0ee8 : bd 46 03 20 d2 ff e8 e0  20
0ef0 : 04 d0 ef a9 0d 20 d2 ff  aa
0ef8 : a2 64 20 c9 ff 20 74 0e  fc
0f00 : a9 fe 20 d2 ff a9 8d 20  4e
0f08 : d2 ff 20 e1 ff d0 03 4c  49
0f10 : e1 0d ee 3d 03 a5 59 49  30
0f18 : 08 85 59 d0 ac 18 a5 5e  32
0f20 : 69 08 85 5e 90 02 e6 5f  2e
0f28 : ad 3d 03 c9 50 f0 03 4c  9f
0f30 : ab 0e a2 64 20 c9 ff a9  bb
0f38 : 0d 20 d2 ff a5 5a 49 19  8e
0f40 : 85 5a a5 5e a6 5c 85 5c  44
0f48 : 86 5e a5 5f a6 5d 85 5d  79
0f50 : 86 5f ee 3c 03 ad 3c 03  5e
0f58 : c9 32 f0 03 4c a6 0e 4c  a2
0f60 : e1 0d a9 00 aa a8 9d 46  25
0f68 : 03 e8 e0 08 d0 f8 aa b9  0b
0f70 : 3e 03 0a 90 0b 48 18 b9  8b
0f78 : 46 03 75 fc 99 46 03 68  e5
0f80 : e8 e0 04 d0 ed a2 00 c8  79
0f88 : c0 08 d0 e3 60 a2 00 8a  2d
0f90 : dd 46 03 d0 05 e8 e0 04  8e
0f98 : d0 f6 60 00 ff 00 ff 00  fb
```

**Listing 1. HI-EDDI und MPS 802**

```
programm : hi-print          0d00 0e22
--------------------------------------

0d00 : a9 7f a2 04 a0 00 20 ba  92
0d08 : ff 20 c0 ff a2 7f 86 67  56
0d10 : 20 c9 ff a9 ff 85 61 a9  4f
0d18 : 07 85 fd a9 1c 85 97 a9  36
0d20 : 00 8d 12 0e a9 28 8d 14  67
0d28 : 0e a2 04 bd b6 0d 20 d2  3a
0d30 : ff ca 10 f7 a9 00 85 63  0f
0d38 : 85 64 ad 12 0e 85 65 a9  93
0d40 : 00 85 fe a5 63 a6 64 a4  bd
0d48 : 65 20 bb 0d a0 00 b1 ac  78
0d50 : a6 fe 9d 16 0e e6 65 e8  1f
0d58 : 86 fe e4 fd d0 e5 a9 00  39
0d60 : a0 07 a6 fd 1e 16 0e 2a  0c
0d68 : ca 10 f9 25 61 09 80 20  fe
0d70 : d2 ff 88 10 ed a5 63 18  30
0d78 : 69 08 85 63 90 02 e6 64  31
0d80 : ce 14 0e d0 b5 a9 0d 20  13
0d88 : d2 ff ad 12 0e 18 69 07  5d
0d90 : 8d 12 0e c6 97 f0 03 4c  28
0d98 : 24 0d a9 04 c5 fd f0 0c  56
0da0 : 85 fd a9 01 85 97 a9 0f  88
0da8 : 85 61 d0 eb a9 0f 20 d2  c9
0db0 : ff a9 7f 4c cc ff 50 00  fb
0db8 : 10 1b 08 85 14 86 15 98  04
0dc0 : 4a 4a 4a aa bd f4 0d 85  da
0dc8 : ad 8a 29 03 aa bd 0e 0e  52
0dd0 : 85 ac 98 29 07 18 65 ac  17
0dd8 : 85 ac a5 14 29 f8 85 63  d6
0de0 : a9 20 05 ad 85 ad 18 a5  02
0de8 : ac 65 63 85 ac a5 ad 65  4a
0df0 : 15 85 ad 60 00 01 02 03  55
0df8 : 05 06 07 08 0a 0b 0c 0d  07
0e00 : 0f 10 11 12 14 15 16 17  0e
0e08 : 19 1a 1b 1c 1e 1f 00 40  d4
0e10 : 80 c0 a1 00 11 00 00 00  6a
0e18 : 00 80 00 00 00 00 00 ff  58
0e20 : 00 ff 02 c8 cc 00 02 90  af
```

**Listing 2. HI-EDDI und MPS 801. Eingabe mit dem MSE.**

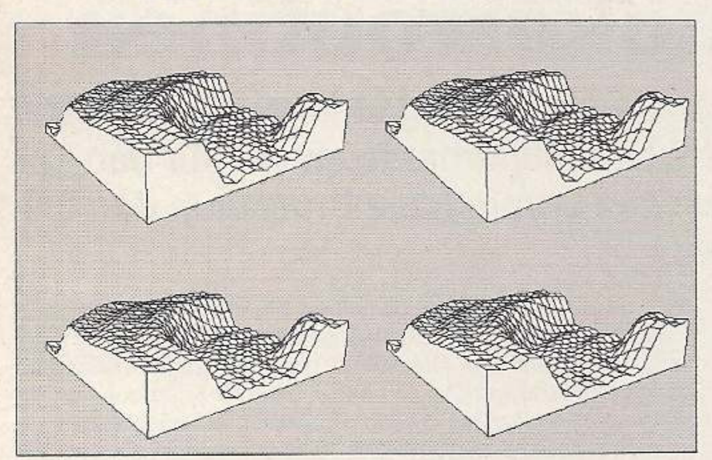



**Bild 1. Eine Grafik mit dem MPS 802 erstellt.**

## HI-EDDI mit der Datasette

Die Anpassung von HI-EDDI an die Datasette war doch nicht ganz so einfach wie zuerst gedacht. Hier sind die notwendigen Informationen. Die folgenden Listingsverweise beziehen sich auf 64'er, Ausgabe 1/85.

a) Am Listing 3 (HI-EDDI) folgende Änderungen vornehmen: In Zeile 50, 110 und 640 LOAD" .. ",8,1 durch LOAD" .. ",1 ersetzen.

```
160  ON A+1 GOSUB 290,250,300,290,290,600,290,290,500,
     510,300:GOTO150
300  POKE8115,1:PRINT...
330  PRINT C$:POKE 780,ASC(C$):H$=W$:W$=""
340  INPUT"{Cursor Down}FILENAME";W$:SYS
     3331:W$=H$:H$=""
350  RETURN
```

Folgende Zeilen weglassen: 200-220, 400

b) Das so modifizierte Listing 3 als erstes auf die Programmkassette aufnehmen.

c) Folgende Befehle im Direktmodus eingeben:

```
NEW
POKE8192,0:POKE43,1:POKE44,32:NEW
```

d) Den DATA-Lader (Listing 2) laden und folgende Änderungen vornehmen:

```
 9   :
10   :
50   D=H(H)*16+H(L):S=S+D:POKEI,D
85   IF A<0 THEN95
90   S=0:A=0:PRINT:NEXT
95   POKE43,0:POKE44,13:POKE45,76:POKE46,31
96   SAVE"HI-EXE",1,1:END
```

e) Programmkassette einlegen (nach Listing 3) und Listing 2 starten. Wenn keine Fehler mehr drin sind, erzeugt der Lader auf der Kassette das Maschinenprogramm HI-EXE.

f) HI-EDDI ist jetzt bereits lauffähig: Programmkassette zurückspulen, Computer kaltstarten, Listing 3 laden und starten. Das Maschinenprogramm wird nun geladen.

g) Die HI-EDDI-Befehle LOAD und SAVE funktionieren ebenfalls, somit kann auch schon das Menü erstellt werden. Die fertige Menütafel (mit Listing 4 verbunden) sollte an dritter Stelle auf die Programmkassette aufgenommen werden.

h) Hier die erforderlichen Änderungen am Listing 1: Zunächst wieder die POKEs wie unter c) eingeben und dann Listing 1 laden und folgende Änderungen vornehmen:

```
720  FOR I=3326 TO 3708
730  READA:IF A<0THEN 730
740  POKEI,A:NEXT
750  POKE43,0:POKE44,13:POKE45,125:POKE46,14
760  SAVE"HI-PRINT",1,1
770  END
```

i) Programmkassette einlegen (an 4. Stelle) und Listing 1 starten. Auf der Kassette wird nun das Maschinenprogramm HI-PRINT erzeugt.

Wenn nun im Programmlauf vor einem Druckvorgang diese Druckroutine geladen wurde, muß die Kassette zurückgespult werden, so daß das Maschinenprogramm HI-EXE nach Beendigung des Druckvorganges wieder geladen werden kann.

(Hans Haberl/gk)


64ER ONLINE

# Samurai

Samurai ist ein Strategiespiel für zwei Personen. Gespielt wird auf einem acht mal acht Felder großem Spielplan. Jeder Spieler besitzt 16 Spielfiguren, darunter der König. Ziel des Spieles ist es, den gegnerischen König zu schlagen oder so viele Spielsteine zu entfernen, daß außer dem König nur eine Figur übrig bleibt. Geschlagen werden Spielsteine, ähnlich wie bei Schach, wenn ein eigener Spielstein auf das Feld des Gegners gesetzt wird. Die Zahl auf jedem Spielstein zeigt an, wie weit der Stein ziehen muß. Nach jedem Zug ändert sich diese Zahl des bewegten Spielsteines. Bei dem König erscheinen nur die Eins und die Zwei, bei den übrigen Steinen Eins bis Vier.

Die Spielsteine dürfen nur waagerecht oder senkrecht bewegt werden. Dabei darf der Spieler einmal rechtwinklig abbiegen. Es dürfen keine Spielsteine übersprungen werden.

Der Computer überwacht die Züge. Sind diese falsch, erscheint auf dem Bildschirm »N. möglich«. Wird ein Spielstein geschlagen, so erscheint »F. geschlagen«. Rechts oben steht immer der Name des Spielers, der am Zug ist.

Man kann die Steine bewegen, indem man die Koordinaten der Anfangs- und Endposition der Steine eingibt. Eine weitere Möglichkeit ist, die Spielsteine durch den Joystick zu steuern.

Mit dem Joystick kann man die Spielsteine folgendermaßen bewegen: am Anfang erscheint ein Pfeil. Der Spieler mit dem roten Stein hat einen weißen Pfeil und umgekehrt. Diesen Pfeil kann man nun beliebig, mit Hilfe des Joysticks, über das Spielfeld bewegen. Hat man den Pfeil auf dem Spielstein, den man aufnehmen will, so muß man auf den Feuerknopf drücken. Nun kann man entsprechend der Zahl auf dem Spielstein fahren. Der Spielstein bleibt aber auf seiner Position. Wenn man den Stein nun absetzen will, so muß man wiederum auf den Feuerknopf drücken. Der Rahmen um den Pfeil verschwindet nun und der Stein wird dort hingesetzt. Wenn man den Stein mit Hilfe der Koordinaten bewegen will, muß man am Anfang (nach der Anleitung) ein »K« drücken. Der jeweilige Spieler muß dann während des Spieles die Zeile und Spalte des Anfangs- und Endpunktes des Spielsteines über Tastatur eingeben. Falls man keinen oder nur einen Joystick besitzt, kann man den fehlenden durch die Tastatur ersetzen.

Man muß folgende Zeilen ändern (Am Beispiel für Spieler 1, Joystickport 1):

```
22210  GET A$
22220  IFA$= "Tastaturbelegung"
       ANDY1-24>50THENY1=Y1-24:
       GOTO22230                     (nach oben)
22221  IFA$= "Tastaturbelegung"
       ANDX1-24>20THENX1=X1-24:
       GOTO22230                     (nach links)
22222  IFA$= "Tastaturbelegung"
       ANDY1+24<230THENY1=Y1+
       24:GOTO22230                  (nach
22223  IFA$= "Tastaturbelegung"       unten)
       ANDX1+24<210THENX1=X1+
       24:GOTO22230                  (nach
22224  IFA$= "Tastaturbelegung"       rechts)
       THENGOTO22300                 (absetzen,
22232  GOTO22210                      aufnehmen)
22314  QR=1:U1=6:POKEV+21,24
```

(Thomas Strigl/rg)

```
1 REM -------------------------                          <245>
2 REM          SAMURAI,                                  <207>
3 REM         WRITTEN BY                                 <090>
4 REM       THOMAS STRIGL                                <052>
5 REM  AUSTR.25,7585 LICHTENAU                           <090>
6 REM -------------------------                          <250>
11 :                                                     <069>
12 :                                                     <070>
13 DIM FE(8,8)                                           <011>
15 SI=54272:FL=SI:FH=SI+1:W=SI+4:A=SI+5:H=SI+6:L=SI+
   24                                                    <101>
20 FA=2:Z=1:S=1:SW=8:SR=8:X1=32:Y1=61:X2=32:Y2=229      <163>
30 POKE 53281,0:POKE 53280,0                             <067>
40 GOSUB 9210                                            <129>
95 :                                                     <153>
96 :                                                     <154>
97 REM * SPIELFELD *                                     <220>
98 PRINT"{GREY 2,2SPACE}1{2SPACE}2{2SPACE}3{2SPACE}4
   {2SPACE}5{2SPACE}6{2SPACE}7{2SPACE}8                  <089>
99 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3R
   IGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                    <070>
100 PRINT"1{RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3RIGHT
    ,3SPACE,3RIGHT,3SPACE,3RIGHT}"                       <120>
101 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <072>
102 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <073>
103 PRINT"2{RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3SPACE
    ,3RIGHT,3SPACE,3RIGHT,3SPACE}"                       <124>
104 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <075>
106 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <077>
107 PRINT"3{RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3RIGHT
    ,3SPACE,3RIGHT,3SPACE,3RIGHT}"                       <129>
108 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <079>
109 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <080>
110 PRINT"4{RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3SPACE
    ,3RIGHT,3SPACE,3RIGHT,3SPACE}"                       <133>
111 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <082>
112 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <083>
113 PRINT"5{RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3RIGHT
    ,3SPACE,3RIGHT,3SPACE,3RIGHT}"                       <137>
114 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <085>
115 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <086>
116 PRINT"6{RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3SPACE
    ,3RIGHT,3SPACE,3RIGHT,3SPACE}"                       <141>
117 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <088>
118 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <089>
119 PRINT"7{RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3RIGHT
    ,3SPACE,3RIGHT,3SPACE,3RIGHT}"                       <145>
120 PRINT"{SPACE,RVSON,GREY 2,3SPACE,3RIGHT,3SPACE,3
    RIGHT,3SPACE,3RIGHT,3SPACE,3RIGHT}"                  <091>
121 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE}"                  <092>
122 PRINT"8{RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3SPACE
    ,3RIGHT,3SPACE,3RIGHT,3SPACE}"                       <149>
123 PRINT"{SPACE,RVSON,GREY 2,3RIGHT,3SPACE,3RIGHT,3
    SPACE,3RIGHT,3SPACE,3RIGHT,3SPACE,HOME}"             <113>
129 PRINT"{HOME}"                                        <113>
131 PRINT TAB(26)" WRITTEN BY"                           <092>
132 PRINT TAB(26)"THOMAS STRIGL"                         <054>
133 PRINT TAB(25)"@@@@@@@@@@@@@@@@"                      <049>
198 :                                                    <000>
199 :                                                    <001>
200 REM *** GRUNDSTELLUNG ***                            <065>
201 B=1065:F=55337:Y=1:V=12:C=9                          <069>
209 FOR Q=1 TO 2                                         <092>
211 IF Y=2 THEN B=1905:F=56177                           <065>
212 FOR X=1 TO V                                         <138>
220 POKE B,160:POKE F,Y                                  <020>
230 B=B+1:F=F+1:NEXT X                                   <094>
240 FOR X=1 TO 3                                         <131>
250 POKE B,160:POKE F,7                                  <016>
260 B=B+1:F=F+1                                          <105>
261 POKE V+5,61                                          <101>
265 NEXT X                                               <228>
266 FOR X=1 TO C                                         <174>
267 POKE B,160:POKE F,Y                                  <068>
268 B=B+1:F=F+1:NEXT X                                   <133>
269 B=B+16:F=F+16                                        <222>
270 FOR X=1 TO 24                                        <213>
271 POKE B,160:POKE F,Y                                  <072>
272 B=B+1:F=F+1:NEXT X                                   <137>
273 B=B+16:F=F+16                                        <226>
274 FOR X=1 TO 24                                        <217>
275 POKE B,160:POKE F,Y                                  <076>
276 B=B+1:F=F+1:NEXT X                                   <141>
277 IF Y=1 THEN A=4:ZA=3:B=1106:F=55378                  <153>
278 IF Y=2 THEN B=1946:F=56218                           <134>
280 FOR X=1 TO A                                         <186>
281 ZU=INT(4*RND(1))+1                                   <121>
282 POKE B,ZU+176:POKE F,Y                               <179>
284 IF Y=1 THEN FE(1,X)=ZU+10                            <136>
285 IF Y=2 THEN FE(8,X)=ZU                               <134>
286 B=B+3:F=F+3:NEXT X                                   <155>
289 ZK=INT(2*RND(1))+1                                   <117>
290 POKE B,ZK+176:POKE F,Y                               <177>
291 B=B+3:F=F+3                                          <140>
293 IF Y=1 THEN FE(1,5)=ZK+30                            <102>
294 IF Y=2 THEN FE(8,4)=ZK+20                            <109>
300 FOR X=1 TO ZA                                        <040>
305 ZU=INT(4*RND(1))+1                                   <145>
306 POKE B,ZU+176:POKE F,Y                               <203>
```

**Listing »Samurai«. Geben Sie dieses Listing bitte mit dem Checksummer 64 ein.**

```
307 IF Y=1 THEN FE(1,X+5)=ZU+10                        <126>
308 IF Y=2 THEN FE(8,X+4)=ZU                           <123>
310 B=B+3:F=F+3:NEXT X                                 <179>
350 A=A-1:ZA=ZA+1:Y=2:C=12:V=9:NEXT Q                  <253>
360 IF QM$="S"THEN GOSUB 22000                         <039>
370 IF QM$="S"THEN GOTO 22100                          <046>
398 :                                                  <201>
399 :                                                  <202>
400 REM *** ZIEHEN ***                                 <223>
401 GOSUB 20100                                        <018>
402 GOSUB 800                                          <184>
420 POKE 214,11:POKE 211,25:SYS 58640                  <056>
430 INPUT"{LIG.BLUE}VON: ZEILE";Z1                     <126>
440 INPUT"{LIG.BLUE,DOWN,29RIGHT}SPALTE";S1            <254>
450 INPUT"{LIG.BLUE,3DOWN,25RIGHT}NACH:ZEILE";Z2       <194>
460 INPUT"{LIG.BLUE,DOWN,29RIGHT}SPALTE";S2            <019>
598 :                                                  <146>
599 :                                                  <147>
600 REM *** FIG. SETZEN ***                             <194>
601 GOTO 7000                                          <171>
620 P=((Z2*3-2)*40+S2*3-2+1024)                        <198>
621 F=((Z2*3-2)*40+S2*3-2+55296)                       <001>
625 FOR X=1 TO 3                                       <006>
630 POKE P,160:POKE F,FA                               <236>
640 P=P+1:F=F+1                                        <002>
641 NEXT X                                             <093>
650 GOSUB 3000                                         <220>
660 P=P+37:F=F+37                                      <136>
661 POKE P,160:POKE F,FA                               <011>
662 P=P+1:F=F+1                                        <024>
670 POKE P,ZU+176:POKE F,FA                            <116>
672 P=P+1:F=F+1                                        <034>
680 POKE P,160:POKE F,FA                               <030>
681 P=P+38:F=F+38                                      <159>
685 FOR X=1 TO 3                                       <066>
690 POKE P,160:POKE F,FA                               <040>
695 P=P+1:F=F+1                                        <057>
696 NEXT X                                             <148>
710 IF FA=2 THEN FE(Z2,S2)=ZU                          <223>
711 IF FA=1 THEN FE(Z2,S2)=ZU+10                       <234>
720 GOTO 6500                                          <038>
721 :                                                  <013>
722 :                                                  <014>
800 REM * NAME *                                       <007>
810 POKE 214,8:POKE 211,25:SYS 58640                   <150>
820 IF SP$=N1$THEN SP$=N2$:PRINT"{YELLOW,3RIGHT}";SP
    $:RETURN                                           <120>
830 IF SP$=N2$THEN SP$=N1$:PRINT"{YELLOW,3RIGHT}";SP
    $:RETURN                                           <130>
2205 POKE V+21,60                                      <049>
2998 :                                                 <251>
2999 :                                                 <252>
3000 REM * ZUFALLSZAHL *                               <246>
3005 ZU=INT(4*RND(1))+1                                <039>
3010 RETURN                                            <091>
3998 :                                                 <231>
3999 :                                                 <232>
4000 REM *** KOENIG ***                                <247>
4010 P=((Z2*3-2)*40+S2*3-2+1024)                       <017>
4020 F=((Z2*3-2)*40+S2*3-2+55296)                      <085>
4030 FOR X=1 TO 3                                      <096>
4040 POKE P,160:POKE F,7                               <251>
4050 P=P+1:F=F+1                                       <097>
4060 NEXT X                                            <197>
4070 GOSUB 5000                                        <071>
4080 P=P+37:F=F+37                                     <241>
4090 POKE P,160:POKE F,FA                              <125>
4100 P=P+1:F=F+1                                       <148>
4110 POKE P,ZK+176:POKE F,FA                           <232>
4120 P=P+1:F=F+1                                       <168>
4130 POKE P,160:POKE F,FA                              <166>
4140 P=P+38:F=F+38                                     <048>
4150 FOR X=1 TO 3                                      <217>
4160 POKE P,160:POKE F,FA                              <196>
4170 P=P+1:F=F+1                                       <218>
4180 NEXT X                                            <062>
4200 IF FA=2 THEN FE(Z2,S2)=ZK+20                      <145>
4210 IF FA=1 THEN FE(Z2,S2)=ZK+30                      <155>
4220 GOTO 6500                                         <224>
4998 :                                                 <211>
4999 :                                                 <212>
5000 REM *** ZUFALL K. ***                             <109>
5010 ZK=INT(2*RND(1))+1                                <248>
5020 RETURN                                            <061>
5998 :                                                 <191>
5999 :                                                 <192>
6500 REM *** FELD LOESCHEN ***                         <116>
6505 R=Z1+S1:FA=0                                      <226>
6510 IF R=2 OR R=4 OR R=6 OR R=8 OR R=10 THEN FA=12    <094>
6520 IF R=12 OR R=14 OR R=16 OR R=18 THEN FA=12        <023>
6530 P=((Z1*3-2)*40+S1*3-2+1024)                       <241>
6540 F=((Z1*3-2)*40+S1*3-2+55296)                      <053>
6545 FOR Q=1 TO 3                                      <054>
6550 FOR X=1 TO 3                                      <066>
6551 POKE P,160:POKE F,FA                              <036>
6552 P=P+1:F=F+1:NEXT X                                <069>
6553 P=P+37:F=F+37:NEXT Q                              <177>
6554 FE(Z1,S1)=0                                       <172>
6555 IF SW=2 OR SR=2 THEN GOTO 9100                    <000>
6560 GOTO 8000                                         <010>
6998 :                                                 <171>
6999 :                                                 <172>
7000 REM *** ZUG RICHTIG? ***                          <061>
7001 GOTO 9000                                         <198>
7010 IF FE(Z1,S1)<10 THEN FA=2                         <069>
```

**Listing »Samurai« (Fortsetzung)**

```
7020 IF FE(Z1,S1)>10 THEN FA=1                         <076>
7030 IF FE(Z1,S1)>20 THEN FA=2                         <088>
7040 IF FE(Z1,S1)>30 THEN FA=1                         <098>
7041 IF FA=1 AND SP$=N2$THEN GOTO 8100                 <086>
7042 IF FA=2 AND SP$=N1$THEN GOTO 8100                 <087>
7050 IF FE(Z2,S2)=0 THEN GOTO 7200                     <036>
7060 IF FE(Z2,S2)<10 THEN SA=2                         <134>
7065 IF FE(Z2,S2)>10 THEN SA=1                         <136>
7070 IF FE(Z2,S2)>20 THEN SA=51                        <195>
7075 IF FE(Z2,S2)>30 THEN SA=52                        <202>
7076 IF FA=1 AND SA=1 THEN GOTO 8100                   <211>
7077 IF FA=1 AND SA=52 THEN GOTO 8100                  <010>
7078 IF FA=2 AND SA=51 THEN GOTO 8100                  <011>
7079 IF FA=2 AND SA=2 THEN GOTO 8100                   <216>
7085 IF SA=51 AND FA=1 THEN GOTO 9100                  <018>
7087 IF SA=52 AND FA=2 THEN GOTO 9100                  <022>
7088 IF FE(Z1,S1)>20 AND SA=1 THEN SW=SW-1:GOTO 4000 <182>
7089 IF FE(Z1,S1)>30 AND SA=2 THEN SR=SR-1:GOTO 4000 <175>
7090 IF FA=2 AND SA=1 THEN SW=SW-1:PRINT TAB(26)"{DO
     WN,RVSON,GREEN}F.GESCHLAGEN":GOSUB 20000          <252>
7100 IF FA=1 AND SA=2 THEN SR=SR-1:PRINT TAB(26)"{DO
     WN,RVSON,GREEN}F.GESCHLAGEN":GOSUB 20000          <252>
7200 IF FE(Z1,S1)>20 THEN GOTO 4000                    <197>
7210 GOTO 620                                          <103>
7998 :                                                 <151>
7999 :                                                 <152>
8000 REM * LOESCHEN *                                  <147>
8009 PRINT"{HOME,4DOWN}"                               <156>
8010 FOR I=1 TO 19                                     <036>
8011 PRINT TAB(25)"{14SPACE}"                          <122>
8012 NEXT                                              <237>
8030 IF SX=1 THEN SX=0:POKE 214,8:POKE 211,25:SYS 58
     640:PRINT"{YELLOW,3RIGHT}";SP$:GOTO 8033          <012>
8031 IF QM$="S"THEN GOTO 22100                         <057>
8032 GOTO 400                                          <156>
8033 IF QM$="S"THEN GOTO 22200                         <060>
8034 GOTO 420                                          <160>
8098 :                                                 <251>
8099 :                                                 <252>
8100 REM * N.MOEGLICH *                                <106>
8110 PRINT TAB(26)"{DOWN,RVSON,GREEN}N.MOEGLICH"       <227>
8120 GOSUB 20000                                       <086>
8130 SX=1:GOTO 8000                                    <250>
8998 :                                                 <131>
8999 :                                                 <132>
9000 REM * ZAHL *                                      <093>
9001 IF S2>8 OR Z2>8 THEN GOTO 8100                    <099>
9002 IF S1>8 OR Z1>8 THEN GOTO 8100                    <098>
9010 IF Z1<Z2 THEN RI=Z2-Z1                            <096>
9011 IF Z1>Z2 THEN RI=Z1-Z2                            <095>
9012 IF S1<S2 THEN RA=S2-S1                            <062>
9013 IF S1>S2 THEN RA=S1-S2                            <061>
9014 IF Z1=Z2 THEN RI=0                                <209>
9015 IF S1=S2 THEN RA=0                                <188>
9024 IF FE(Z1,S1)<10 THEN GOTO 9030                    <021>
9025 IF FE(Z1,S1)<20 THEN GOTO 9040                    <024>
9026 IF FE(Z1,S1)<30 THEN GOTO 9050                    <027>
9027 IF FE(Z1,S1)<40 THEN GOTO 9060                    <030>
9030 REM STEIN ROT                                     <112>
9031 IF RI+RA<>FE(Z1,S1)THEN GOTO 8100                 <065>
9032 GOTO 10000                                        <229>
9040 REM STEIN WEISS                                   <016>
9041 IF RI+RA<>FE(Z1,S1)-10 THEN GOTO 8100             <087>
9042 GOTO 10000                                        <239>
9050 REM KOENIG ROT                                    <190>
9051 IF RI+RA<>FE(Z1.S1)-20 THEN GOTO 8100             <098>
9052 GOTO 10000                                        <249>
9060 REM KOENIG WEISS                                  <094>
9061 IF RI+RA<>FE(Z1,S1)-30 THEN GOTO 8100             <109>
9062 GOTO 10000                                        <003>
9098 :                                                 <231>
9099 :                                                 <232>
9100 REM * GEWONNEN *                                  <243>
9101 POKE V+21,0                                       <006>
9105 PRINT"{CLR}"                                      <036>
9110 PRINT"{2DOWN,4RIGHT}GLUECKWUNSCH !!!"             <034>
9111 PRINT"{4DOWN,4RIGHT}";SP$;" HAT GEWONNEN"         <014>
9112 GOSUB 20000                                       <058>
9113 INPUT"{4RIGHT,3DOWN}NOCH EIN SPIEL (J/N)";XY$     <213>
9114 IF XY$="N"THEN END                                <136>
9115 IF XY$="J"THEN CLR:GOTO 1                         <149>
9116 :                                                 <249>
9117 :                                                 <250>
9118 REM *** ANLEITUNG ***                             <243>
9119 POKE 56322,255                                    <035>
9210 PRINT CHR$(14)"{CLR}                              <232>
9211 PRINT"{YELLOW,2SPACE,RVSON,SPACE}S A M U R A I{
     SPACE,RVOFF},WRITTEN BY THOMAS"                   <015>
9212 PRINT"{29SPACE}STRIGL                             <047>
9213 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"     <249>
9398 :                                                 <020>
9399 :                                                 <021>
9400 PRINT"{3DOWN,YELLOW,SPACE}ZIEL DES{SHIFT-SPACE}
     SPIELS:                                           <034>
9401 PRINT"                                            <152>
9402 PRINT"{GREEN,SPACE}BEI SAMURAI GEHT ES DARUM ,D
     IE HAUPT-                                         <255>
9403 PRINT" FIGUR DES GEGENSPIELERS,SEINEN KOENIG,     <151>
9404 PRINT" ZU BESIEGEN ODER SEINE SPIELSTEINE ZU      <158>
9405 PRINT" SCHLAGEN, DASS NUR NOCH DER KOENIG UND     <084>
9406 PRINT" EIN WEITERER STEIN UEBRIGBLEIBEN.          <192>
9407 PRINT"{ORANGE,4SPACE}DIE ZAHL AUF JEDEM SPIELST
     EIN GIBT                                          <007>
9408 PRINT" AN,WIE WEIT DER STEIN ZIEHEN MUSS.         <143>
9409 PRINT" DAS BESONDERE IST, DASS SICH DIE ZAHL      <254>
```

```
9410 PRINT" BEI JEDEM ZUG AENDERT.MAN MUSS DESHALB   <020>
9411 PRINT" GLUECK UND UEBERRASCHUNGEN IN SEINE       <176>
9412 PRINT" STRATEGIE EINBEZIEHEN !                   <018>
9420 PRINT"{YELLOW,DOWN,11SPACE}>>>> TASTE <<<<       <195>
9422 GET A$:IF A$=""THEN GOTO 9422                    <025>
9450 PRINT"{CLR,LIG.GREEN,3DOWN,SPACE}WIE DIE SPIELS
     TEINE GEZOGEN WERDEN:                            <050>
9451 PRINT"                                           <202>
9452 PRINT"{LIG.BLUE,SPACE}DIE ZAHL, DIE IM SPIELSTE
     IN SICHTBAR                                      <202>
9453 PRINT" IST, ZEIGT AN, WIE VIELE FELDER WEIT      <235>
9454 PRINT" DER STEIN ZIEHEN MUSS. BEIM KOENIG        <190>
9456 PRINT" ERSCHEINEN NUR DIE ZAHLEN 1 UND 2 ,       <050>
9457 PRINT" BEI DEN UEBRIGEN STEINEN DIE ZAHLEN       <114>
9458 PRINT" 1 BIS 4.                                  <066>
9459 PRINT"{6SPACE}DIE SPIELSTEINE DUERFEN NUR        <231>
9460 PRINT" WAAGRECHT UND SENKRECHT GEZOGEN WERDEN"   <237>
9461 PRINT" WERDEN, NICHT SCHRAEG.                    <102>
9462 PRINT" BEI JEDEM ZUG DARF MAN NUR EINMAL         <036>
9463 PRINT" RECHTWINKLIG ABBIEGEN.                    <188>
9464 PRINT" ES DUERFEN KEINE SPIELSTEINE              <041>
9465 PRINT" UEBERSPRUNGEN WERDEN !!!!!!. SPIELER 1    <059>
9466 PRINT" FAENGT MIT WEISS AN.                      <192>
9468 PRINT"{DOWN,SPACE}VIEL SPASS !!!                 <137>
9470 INPUT"{2DOWN,SPACE}NAME SPIELER 1";N1$           <081>
9471 INPUT" NAME SPIELER 2";N2$                       <050>
9472 SP$=N2$                                          <066>
9473 PRINT"{CLR}"                                     <150>
9474 PRINT:PRINT:PRINT                                <102>
9475 PRINT"{YELLOW,5SPACE}HALT !!!! ":PRINT           <163>
9476 PRINT"{YELLOW,5SPACE}ES GIBT 2 MOEGLICHKEITEN U
     M "                                              <062>
9477 PRINT"{5SPACE}DIE FIGUREN ZU BEWEGEN.":PRINT:PR
     INT                                              <233>
9478 PRINT"{5SPACE}K - DURCH EINGABE VON KOORDINATEN
     "                                                <018>
9479 PRINT                                            <197>
9480 PRINT"{5SPACE}S - DURCH BEWEGEN U. STEUERN UEBE
     R"                                               <025>
9481 PRINT"{9SPACE}JOYSTICK"                          <251>
9482 PRINT:PRINT:PRINT                                <110>
9483 INPUT"{5SPACE}BUCHSTABE";QM$                     <007>
9980 PRINT CHR$(142)"{CLR}":RETURN                    <009>
9981 :                                                <093>
9982 :                                                <094>
10000 REM * FIG. UEBERSPRUNGEN? *                     <066>
10001 IF Z1=Z2 THEN GOTO 10088                        <189>
10002 IF S1=S2 THEN GOTO 10094                        <173>
10010 IF Z1>Z2 THEN GOTO 10050                        <186>
10020 IF Z2>Z1 THEN GOTO 10070                        <198>
10050 :                                               <163>
10051 FOR I=Z1-1 TO Z2 STEP-1                         <002>
10052 IF FE(I,S1)<>0 THEN GOTO 10059                  <142>
10053 NEXT                                            <238>
10054 FOR I=S1 TO S2-1                                <114>
10055 IF S1>S2 THEN FOR I=S2+1 TO S1                  <094>
10056 IF FE(Z2,I)<>0 THEN GOTO 10059                  <154>
10057 NEXT:GOTO 7010                                  <125>
10059 FOR I=S1+1 TO S2                                <118>
10060 IF S1>S2 THEN FOR I=S1-1 TO S2 STEP-1           <233>
10061 IF FE(Z1,I)<>0 THEN GOTO 8100                   <104>
10062 NEXT                                            <247>
10063 FOR I=Z2+1 TO Z1                                <136>
10064 IF FE(I,S2)<>0 THEN GOTO 8100                   <101>
10065 NEXT:GOTO 7010                                  <133>
10070 :                                               <183>
10071 FOR I=Z1+1 TO Z2                                <144>
10072 IF FE(I,S1)<>0 THEN GOTO 10079                  <164>
10073 NEXT                                            <002>
10074 FOR I=S2+1 TO S1                                <133>
10075 IF S1<S2 THEN FOR I=S1+1 TO S2-1                <080>
10076 IF FE(Z2,I)<>0 THEN GOTO 10079                  <176>
10077 NEXT:GOTO 7010                                  <145>
10079 FOR I=S2 TO S1-1                                <139>
10080 IF S1<S2 THEN FOR I=S1+1 TO S2                  <121>
10081 IF FE(Z1,I)<>0 THEN GOTO 8100                   <124>
10082 NEXT                                            <011>
10083 FOR I=Z1 TO Z2-1                                <157>
10084 IF FE(I,S2)<>0 THEN GOTO 8100                   <121>
10085 :                                               <198>
10088 IF S1-S2=1 THEN GOTO 7010                       <169>
10089 IF S2-S1=1 THEN GOTO 7010                       <170>
10090 FOR I=S1+1 TO S2-1                              <113>
10091 IF S1>S2 THEN FOR I=S1-1 TO S2+1 STEP-1         <227>
10092 IF FE(Z1,I)<>0 THEN GOTO 8100                   <135>
10093 NEXT:GOTO 7010                                  <161>
10094 IF Z1-Z2=1 THEN GOTO 7010                       <189>
10095 IF Z2-Z1=1 THEN GOTO 7010                       <190>
10096 FOR I=Z1+1 TO Z2-1                              <133>
10097 IF Z1>Z2 THEN FOR I=Z1-1 TO Z2+1 STEP-1         <005>
10098 IF FE(I,S1)<>0 THEN GOTO 8100                   <134>
10099 NEXT:GOTO 7010                                  <167>
10100 :                                               <213>
10101 :                                               <214>
20000 REM *** TOENE ***                               <084>
20010 POKE L,15                                       <237>
20020 POKE A,16+9                                     <208>
20040 POKE H,4*16+4                                   <198>
20041 FOR Q=1 TO 5                                    <037>
20050 POKE FH,29:POKE FL,69                           <090>
20060 POKE W,33                                       <042>
20070 FOR T=1 TO 40:NEXT T                            <132>
20071 POKE FH,17:POKE FL,103                          <145>
20072 POKE W,33                                       <054>
20073 FOR T=1 TO 40:NEXT T                            <135>
```

**Listing »Samurai« (Schluß)**

```
20074 NEXT Q                                          <139>
20080 POKE W,0:POKE A,0:POKE H,0                      <235>
20082 RETURN                                          <078>
20100 :                                               <012>
20110 POKE L,15                                       <081>
20120 POKE A,16+9                                     <052>
20140 POKE H,4*16+4                                   <042>
20150 POKE FH,29:POKE FL,69                           <190>
20160 POKE W,33                                       <142>
20170 FOR I=1 TO 500:NEXT                             <186>
20180 POKE W,0:POKE A,0:POKE H,0                      <079>
20190 RETURN                                          <186>
20998 :                                               <146>
20999 :                                               <147>
22000 REM ** DATEN F. SPRITES **                      <134>
22001 :                                               <128>
22002 SP$=N2$                                         <100>
22004 V=53248                                         <087>
22005 POKE V+21,60                                    <214>
22006 POKE 2042,13:POKE 2043,14                       <101>
22007 POKE 2044,13:POKE 2045,14                       <106>
22009 :                                               <136>
22010 REM SPRITE 2                                    <231>
22012 FOR N=0 TO 62:READ Q:POKE 832+N,Q:NEXT          <191>
22014 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0                <014>
22015 DATA 0,0,0,0,0,0,0,0,0,0,0                      <251>
22016 DATA 0,0,0,0,0,0,0,0,0,255,0                    <105>
22017 DATA 1,255,128,3,255,192,7,255,224              <120>
22018 DATA 15,255,240,31,255,248                      <237>
22019 DATA 63,255,252,127,255,254,255,255,255         <127>
22020 POKE V+39+5,7                                   <159>
22021 :                                               <149>
22050 REM SPRITE 3                                    <017>
22051 FOR N=0 TO 62:READ Q:POKE 896+N,Q:NEXT          <241>
22052 DATA 255,255,255,128,0,1,128,0,1,128,0,1,128,0
      ,1                                              <041>
22053 DATA 128,0,1,128,0,1,128,0,1,128,0,1            <210>
22054 DATA 128,0,1,128,0,1,128,0,1,128,255,1          <063>
22055 DATA 129,255,129,131,255,193,135,255,225        <208>
22056 DATA 143,255,241,159,255,249                    <130>
22057 DATA 191,255,253,255,255,255,255,255,255        <219>
22058 POKE V+4,32:POKE V+5,61                         <117>
22059 POKE V+8,32:POKE V+9,230                        <172>
22060 POKE V+41,2:POKE V+42,2:POKE V+43,1:POKE V+44,
      1                                               <156>
22070 RETURN                                          <026>
22071 :                                               <199>
22072 :                                               <200>
22100 REM *** SPRITES BEWEGEN ***                     <092>
22101 GOSUB 800                                       <208>
22200 GOSUB 20100:IF SP$=N2$THEN GOTO 22400           <152>
22201 REM ** SPIELER 1 **                             <139>
22202 U1=4:QR=0                                       <059>
22210 J=PEEK(56321)                                   <040>
22220 IF(J AND 1)=0 AND Y1-24>50 THEN Y1=Y1-24:GOTO
      22230                                           <164>
22221 IF(J AND 4)=0 AND X1-24>20 THEN X1=X1-24:GOTO
      22230                                           <162>
22222 IF(J AND 2)=0 AND Y1+24<230 THEN Y1=Y1+24:GOTO
       22230                                          <215>
22223 IF(J AND 8)=0 AND X1+24<210 THEN X1=X1+24:GOTO
       22230                                          <217>
22224 IF(J AND 16)=0 THEN GOTO 22300                  <107>
22225 GOTO 22210                                      <167>
22230 POKE V+U1,X1:POKE V+U1+1,Y1                     <230>
22231 FOR I=1 TO 80:NEXT I:GOTO 22210                 <165>
22300 IF QR<>0 THEN GOTO 22350                        <097>
22310 Z1=((Y1-61)/24)+1                               <230>
22312 S1=((X1-32)/24)+1                               <222>
22313 IF FE(Z1,S1)=0 THEN GOTO 22210                  <043>
22314 QR=1:U1=6:POKE V+21,24:FOR I=1 TO 160:NEXT      <083>
22315 POKE V+U1,X1:POKE V+U1+1,Y1                     <060>
22316 GOTO 22210                                      <003>
22350 Z2=((Y1-61)/24)+1                               <015>
22352 S2=((X1-32)/24)+1                               <007>
22354 U1=4:POKE V+21,20:POKE V+U1,X1:POKE V+U1+1,Y1   <203>
22356 GOTO 600                                        <202>
22357 :                                               <230>
22400 REM ** SPIELER 2 **                             <084>
22402 U2=8:QR=0                                       <009>
22403 POKE 56322,255                                  <059>
22410 J=PEEK(56320)                                   <240>
22420 IF(J AND 1)=0 AND Y2-24>50 THEN Y2=Y2-24:GOTO
      22500                                           <112>
22421 IF(J AND 4)=0 AND X2-24>20 THEN X2=X2-24:GOTO
      22500                                           <110>
22422 IF(J AND 2)=0 AND Y2+24<230 THEN Y2=Y2+24:GOTO
       22500                                          <163>
22423 IF(J AND 8)=0 AND X2+24<210 THEN X2=X2+24:GOTO
       22500                                          <165>
22424 IF(J AND 16)=0 THEN GOTO 22520                  <056>
22425 GOTO 22410                                      <114>
22500 POKE V+U2,X2:POKE V+U2+1,Y2                     <249>
22510 I=0:FOR I=1 TO 80:NEXT I:GOTO 22410             <036>
22520 IF QR<>0 THEN GOTO 22600                        <059>
22525 Z1=((Y2-61)/24)+1                               <190>
22530 S1=((X2-32)/24)+1                               <186>
22531 IF FE(Z1,S1)=0 THEN GOTO 22410                  <008>
22535 QR=1:U2=10:POKE V+21,36:FOR I=1 TO 160:NEXT     <096>
22536 POKE V+U2,X2:POKE V+U2+1,Y2                     <030>
22537 GOTO 22410                                      <227>
22600 Z2=((Y2-61)/24)+1                               <011>
22601 S2=((X2-32)/24)+1                               <002>
22602 U2=8:POKE V+21,20:POKE V+U2,X2:POKE V+U2+1,Y2   <205>
22603 GOTO 600                                        <194>
```


64ER ONLINE

# Prost mit dem C 64

**Dieses Programm macht Ihren C 64 zum elektronischen Steuermann. Mit dem User-Port können Sie bis zu acht Geräte ein- und ausschalten.**

Der Name Pro.St steht für programmierbare Steuerung; das Programm erlaubt die Steuerung von acht Geräten, die über entsprechende Treiberschaltungen am User-Port des C 64 angeschlossen sind. Für den Steuerungsablauf stehen Ihnen etwa 30 000 Einzelschritte zur Verfügung. Jeder Schritt wird als ein Byte im verbleibenden Basic-Speicher abgelegt. Zur Bearbeitung des Programmablaufs hat Pro.St einen eigenen Editor, mit dem Abläufe erstellt, geändert, angezeigt, ausgeführt, gespeichert und geladen werden.

Nach dem Starten des Programms wird in Zeile 9 das User-Port-Display von Jan Legenhausen, 64'er Ausgabe 8/84, geladen. Sollten Sie dieses Programm (noch) nicht haben, sei gesagt, daß Pro.St auch ohne diese Anzeige läuft, dann aber an eine Autofahrt mit schwarzen Scheiben erinnert: Man weiß nie, woran man ist. Haben Sie das Programm von Jan Legenhausen nur als Basic-Lader, starten Sie den Lader und laden dieses Programm manuell. Vergessen Sie dann nicht, Zeile 9 von Pro.St zu löschen.

Pro.St meldet sich nach dem Start mit seinem Hauptmenü, das sieben Wahlmöglichkeiten hat:

**1:** Eingabe eines Steuerprogramms über die Tastatur. Es werden für jeden Kanal des User-Ports die Schaltzustände (an/aus) binär eingegeben. Tippfehler können Sie mit DELete löschen. Korrekte Eingaben schließen mit RETURN ab. Das Programm wandelt die Eingabe in eine Dezimalzahl und POKEt sie in den Programmspeicher, der ab Adresse 10240 im Basic-RAM beginnt. Beendet wird der Eingabemodus durch Eingabe von E und bei vollem Programmspeicher (RAM-Ende bei 40960).

**2:** Schaltet in den Edit-Modus. In einem Untermenü können Sie bestimmen, ob ein Schritt gelöscht (1), eingefügt (2) oder ersetzt (3) werden soll. Um den entsprechenden Schritt zu ändern, muß dessen Nummer eingetippt werden. Die letzte Wahlmöglichkeit (4) zeichnet eine grafische Darstellung des Programmablaufs. Die Schaltzustände am User-Port werden dazu in einem Diagramm gegen die Zeit aufgetragen, indem bis zu 35 Schritte aus dem Speicher geholt, ins Binärsystem umgerechnet und als High-Low-Diagramm auf den Bildschirm gedruckt werden. Mit den V- und Z-Tasten kann dabei im Programmablaufplan geblättert werden.

| Variable | Bedeutung |
|---|---|
| T$ | Tastaturabfrage |
| BZ$(8) | Diagrammzeile |
| DS$ | Diskstatus |
| NA$ | Filename |
| AN | Anfang des Steuerprogrammspeichers-1 |
| BI(8) | Binärwert |
| D | Dezimalwert |
| HI(8) | Umwandlungshilfe |
| MA | Anzahl der Steuerschritte |
| SS | Aktueller Schritt |
| PO | Portadresse |
| SV | Flag für Speicherbelegung |
| UF | Unterprogramm-Flag |

**Liste der wichtigsten Variablen**

**3:** Führt ins Disketten-Unterprogramm. Hier können erstellte Programme gespeichert (1), geladen (2) und Directories gezeigt werden.

**4:** Löscht den Programmspeicher von Pro.St.

**5:** Startet das Programm im Speicher. Zuvor können Sie noch zwischen Freilauf (1) und Einzelschrittverarbeitung (2) wählen. Bei Freilauf müssen Sie die Zeit zwischen zwei Schritten angeben. Bei Einzelschritt können Sie mit der CTRL-Taste den nächsten Schritt starten. In beiden Fällen bestimmen Sie zusätzlich die Anzahl der Programmdurchläufe.

**6:** Dient zum direkten Testen der angeschlossenen Peripherie. Mit den Tasten 1 bis 8 kann ein Gerät, das am entsprechenden User-Port-Kanal angeschlossen ist, geschaltet werden.

**7:** Beendet das Programm. Nach einer Sicherheitsabfrage wird der User-Port wieder in den Anfangszustand versetzt.

**Hinweis**

Da das Programm in Basic geschrieben ist, kann es nicht für zeitkritische Fälle verwendet werden. Wegen der Ungenauigkeit der internen Uhr des C 64 bedarf es einer Zeitkorrektur. Sie sollten den Korrekturfaktor in Zeile 2065 mit einer Stoppuhr nachprüfen. Der Korrekturfaktor wird mit Ihrer Sekundenangabe (ZD) für einen Schaltschritt multipliziert und dann ständig mit der internen Uhr (TI) verglichen. Da TI im 1/60-Sekundentakt zählt, ist der Standard für den Korrekturfaktor 60.

(Michael Anton/hm)

```
0 REM **********************************
     *                                *    <119>
1 REM *      P R O.S T  6 4  (3)      *
     *                                *    <236>
2 REM *         1984  BY              *
     *        MICHAEL ANTON           *    <029>
3 REM *       HERM.ESSIG STR 106      *
     *       7141 SCHWIEBERDINGEN     *    <189>
4 REM *        (07150)/31576          *
     *                                *    <190>
5 REM **********************************   <040>
7 REM "ZEILE 9 U.10 WEGLASSEN, WENN SIE DA
  S USERPORT-DISPLAY AUS DER 64'ER"        <233>
8 REM "NICHT ALS MASCHINENPROGRAMM NACHLAD
  EN KOENNEN"                              <184>
9 IF SV=0 THEN SV=1:LOAD"UPD.49309",8,1    <068>
10 SYS 49309                               <177>
11 :                                       <069>
12 POKE 56,40:CLR                          <116>
13 :                                       <071>
14 :                                       <072>
15 REM *INIT*                              <038>
16 :                                       <074>
20 AN=10240:SV=0:UF=0:BS$(0)="R":BS$(1)="E
   ":PO=56577:POKE 56579,255:POKE PO,0     <066>
30 POKE 53280,15:POKE 53281,8:PRINT"{CLR,C
   YAN}"                                   <202>
40 FOR L=1 TO 8:READ HI(L):NEXT L          <171>
41 DATA 8,7,6,5,4,3,2,1                    <132>
50 GOTO 10000                              <172>
70 :                                       <128>
75 :                                       <133>
80 REM *LOESCH1*                           <034>
81 INPUT"{CLR,7DOWN,3RIGHT}SICHER (J/N)";A
   $:IF A$<>"J"THEN RETURN                 <066>
85 MA=0:SV=0:RETURN                        <082>
87 :                                       <145>
90 REM *LOESCH2*                           <045>
92 FOR LL=AN+SS TO AN+MA+1:POKE LL,PEEK(LL
   +1):NEXT LL:MA=MA-1:RETURN              <103>
93 :                                       <151>
95 REM *EINF*                              <100>
97 FOR LL=AN+MA+1 TO AN+LX STEP-1:POKE LL+
   1,PEEK(LL):NEXT LL:MA=MA+1:RETURN       <238>
98 :                                       <156>
99 :                                       <157>
100 REM *EINGABE*                          <050>
101 :                                      <159>
102 IF SV=0 THEN 105                       <185>
```

**Listing zu Pro.St**

```
103 INPUT"{CLR,5DOWN,RIGHT}ALTES PROGRAMM
    UEBERSCHREIBEN (J/N)";A$                <209>
104 IF A$<>"J"THEN RETURN                   <127>
105 UF=0                                    <230>
110 PRINT"{CLR,2DOWN,3RIGHT}EINGABE{3DOWN}
    ":PRINT"{RVSON}0{RVOFF,2SPACE,RVSON}1{
    RVOFF,2SPACE,RVSON}DEL{RVOFF,3SPACE,RV
    SON}RETURN{RVOFF,3SPACE,RVSON}E{RVOFF}
    NDE"                                    <242>
120 GOSUB 6000                              <203>
130 :                                       <188>
160 REM *EINSCH*                            <061>
170 LX=1                                    <049>
180 PRINT"{DOWN}SCHRITT "LX;TAB(15);"87654
    321":PRINT TAB(15);                     <055>
190 :                                       <248>
200 REM *BIN-IN*                            <072>
210 FOR LB=1 TO 9                           <161>
220 GET T$:IF(T$="0"OR T$="1")AND LB<9 THE
    N 260                                   <096>
225 IF T$=CHR$(13)AND LB=9 THEN 290         <124>
230 IF T$=CHR$(20)THEN LB=LB-1:PRINT"{LEFT
    }";:GOTO 220                            <172>
240 IF T$="E"THEN 340                       <108>
250 GOTO 220                                <023>
260 BI(LB)=VAL(T$):PRINT T$;:NEXT LB        <127>
270 :                                       <073>
290 REM *BIN-DEZ*                           <239>
295 D=0:H=128                               <029>
300 FOR LB=1 TO 8:D=D+BI(LB)*H:H=H/2:NEXT
    LB                                      <028>
305 IF UF=1 THEN RETURN                     <112>
310 PRINT"{2RIGHT}";D:POKE AN+LX,D          <235>
315 IF UF=2 THEN RETURN                     <123>
320 LX=LX+1:IF LX+AN<40960 THEN 180         <174>
330 PRINT"{2DOWN,2RIGHT}SPEICHERPLATZ ZUEN
    DE!!"                                   <111>
340 PRINT:PRINT"EINGABEENDE BEI SCHRITT ";
    LX-1:MA=LX-1:SV=1:GOSUB 6000            <005>
350 RETURN                                  <237>
400 :                                       <203>
450 :                                       <253>
500 REM *EDITIEREN*                         <113>
502 :                                       <049>
505 IF SV=0 THEN PRINT"{CLR,5DOWN,5RIGHT}S
    PEICHER LEER":GOTO 6000                 <017>
510 PRINT"{CLR,5DOWN}1 ← SCHRITT LOESCHEN"
    :PRINT"{DOWN}2 ← SCHRITT EINFUEGEN"     <054>
520 PRINT"{DOWN}3 ← SCHRITT ERSETZEN":PRIN
    T"{DOWN}4 ← VERLAUFSDIAGRAMM":PRINT"{D
    OWN}5 ← MENU"                           <113>
530 GET T$:IF T$=""THEN 530                 <159>
535 WA=VAL(T$):IF WA<1 OR WA>5 THEN 530     <159>
540 ON WA GOTO 560,580,587,600,550          <140>
545 GOTO 500                                <065>
550 RETURN                                  <182>
555 :                                       <103>
560 REM *LOESCHEN*                          <102>
565 INPUT"{CLR,5DOWN}ZU LOESCHENDER SCHRIT
    T: ";SS:GOSUB 90                        <047>
570 PRINT"{3DOWN,5RIGHT}OK":GOSUB 6000:GOT
    O 500                                   <092>
575 :                                       <123>
580 REM *EINFUEGEN*                         <191>
585 INPUT"{CLR,5DOWN,5RIGHT}EINFUEGEN BEI
    SCHRITT: ";LX:GOSUB 95:GOTO 590         <195>
587 INPUT"{CLR,5DOWN,5RIGHT}ZU ERETZENDER
    SCHRITT ";LX                            <054>
590 PRINT"{DOWN}NEUEN SCHRITT EINGEBEN:{2D
    OWN}":UF=2:GOSUB 180                    <140>
595 GOSUB 6000:GOTO 500                     <000>
597 :                                       <145>
600 REM *DIAGRAMM*                          <127>
602 :                                       <150>
605 MS$="QEEEEEEEEEEEEEEEEEEEEEEEEEEE
    EW":SS=1                                <011>
610 KO$="{3SPACE,RVSON}V{RVOFF}OR{4SPACE,R
    VSON}Z{RVOFF}URUECK{5SPACE,RVSON}A{RVO
    FF}NFANG{5SPACE,RVSON}E{RVOFF}NDE"      <143>
615 FOR L=1 TO 8:BZ$(L)="":NEXT L:HZ=1      <073>
617 POKE 211,15:POKE 214,24:SYS 58640:PRIN
    T"BITTE WARTEN!"                        <130>
620 D=PEEK(AN+SS):H=128                     <037>
630 FOR L=1 TO 8:BI=INT(D/H)                <186>
650 BZ$(HI(L))=BZ$(HI(L))+BS$(BI)           <251>
670 D=D-H*BI:H=H/2:NEXT L                   <103>
680 HZ=HZ+1:SS=SS+1:IF SS>MA THEN 700       <118>
690 IF HZ<=35 THEN 620                      <237>
700 PRINT"{CLR,DOWN}":FOR L=1 TO 8:PRINT L
    ;BZ$(L):PRINT:NEXT L:PRINT              <106>
710 PRINT TAB(3);"{UP}";MS$:PRINT:PRINT TA
    B(2);SS-HZ+1:PRINT:PRINT KO$            <128>
720 GET T$:IF T$=""THEN 720                 <094>
725 IF T$="V"AND SS<MA THEN 615             <255>
730 ZF=0:IF T$="Z"AND SS>35 THEN SS=SS-70:
    ZF=1:IF SS<1 THEN SS=1                  <076>
735 IF T$="Z"AND ZF=1 THEN 615              <169>
740 IF T$="A"THEN SS=1:GOTO 615             <175>
745 IF T$="E"THEN 500                       <101>
750 GOTO 720                                <018>
1000 :                                      <037>
1500 :                                      <027>
2000 REM *START*                            <072>
2002 :                                      <019>
2003 IF SV=0 THEN PRINT"{CLR,5DOWN,5RIGHT}
     SPEICHER LEER":GOTO 6000               <241>
2005 PRINT"{CLR,3DOWN,6RIGHT}LAUF":PRINT"{
     3DOWN}1 ← FREILAUF"                    <125>
2010 PRINT"{DOWN}2 ← EINZELSCHRITT":PRINT"
     {3DOWN,5RIGHT}WAHL?"                   <142>
2015 WAIT 203,63:GET T$                     <245>
2020 IF T$="1"THEN 2040                     <130>
2025 IF T$="2"THEN 2200                     <134>
2030 GOTO 2015                              <070>
2040 PRINT"{CLR,6DOWN,3RIGHT}FREILAUF":PRI
     NT"{3DOWN}1 ← DAUERLAUF"               <237>
2045 PRINT"{DOWN}2 ← X*DURCHLAUFEN":PRINT"
     {3DOWN,5RIGHT}WAHL?"                   <124>
2046 WL=0                                   <138>
2050 WAIT 203,63:GET T$                     <025>
2055 IF T$="1"THEN WL=1                     <102>
2060 IF T$="2"THEN WL=2                     <109>
2062 IF WL=0 THEN 2050                      <148>
2064 :                                      <082>
2065 INPUT "{CLR,4DOWN,3RIGHT}ZYKLUSDAUER
     IN SEKUNDEN ";ZD:ZD=ZD*55              <126>
2070 IF WL=2 THEN INPUT"{2DOWN,3RIGHT}DURC
     HLAEUFE ";DL                           <012>
2075 INPUT"{2DOWN,3RIGHT}START BEI SCHRITT
     ";SS:LL=0:PRINT"{3DOWN,5RIGHT}START
     = < TASTE >"                           <026>
2078 GET T$:IF T$=""THEN 2078               <234>
2080 L=AN+SS                                <229>
2090 POKE PO,PEEK(L):TI$="000000"           <004>
2110 IF TI<ZD THEN 2110                     <042>
2120 IF PEEK(653)=1 THEN K$="BREAK":LL=LL+
     1:SS=SS+1:GOTO 2160                    <118>
2130 SS=SS+1:IF SS<=MA THEN 2080            <002>
2135 LL=LL+1:IF LL<DL THEN SS=1:GOTO 2080   <121>
2140 IF WL=1 THEN SS=1:GOTO 2080            <050>
2150 K$="ENDE"                              <239>
2160 PRINT"{3DOWN}";K$;" BEI DURCHLAUF";LL
     ;" SCHRITT";SS-1:GOTO 6000             <216>
2180 :                                      <198>
2200 REM *EINZEL*                           <074>
2210 INPUT"{CLR,3DOWN,3RIGHT}DURCHLAEUFE "
     ;DL                                    <131>
2220 INPUT"{2DOWN,3RIGHT}START BEI ";SS:PR
     INT"{4DOWN,5RIGHT}START = < TASTE >":
     LL=0                                   <155>
2230 GET T$:IF T$=""THEN 2230               <120>
2240 L=AN+SS:POKE PO,PEEK(L)                <160>
2250 IF PEEK(653)=1 THEN K$="BREAK":LL=LL+
     1:GOTO 2160                            <229>
2260 IF PEEK(653)<>4 THEN 2250              <032>
2265 FOR D=1 TO 50:NEXT D                   <146>
2270 SS=SS+1:IF SS<=MA THEN 2240            <140>
2280 LL=LL+1:IF LL<DL THEN SS=1:GOTO 2240   <008>
2290 K$="ENDE":GOTO 2160                    <007>
2400 :                                      <163>
2500 :                                      <007>
3000 REM *SAVEN*                            <035>
3010 PRINT"{CLR,7DOWN,12RIGHT}ABSAVEN"      <016>
3020 INPUT"{2DOWN,5RIGHT}FILENAME:";NA$     <188>
3030 OPEN 1,8,2,NA$+",S,W":GOSUB 5000       <206>
```

**Listing zu Pro.St (Fortsetzung)**

```
3035 PRINT"{DOWN,5RIGHT}";DS$:IF D1>19 THE
     N 3085                                  <039>
3040 LL=MA+1:PRINT#1,LL                      <101>
3050 FOR L=1 TO LL:PRINT#1,PEEK(AN+L):NEXT
     L                                       <176>
3085 CLOSE 1:CLOSE 15:IF D1>19 THEN 3020     <235>
3090 PRINT"{DOWN,5RIGHT}FILE GESPEICHERT":
     GOSUB 6000                              <125>
3100 RETURN                                  <182>
3200 :                                       <198>
3300 :                                       <042>
4000 REM *EINLESEN*                          <229>
4010 PRINT"{CLR,5DOWN,9RIGHT}EINLESEN"       <214>
4012 IF SV=0 THEN 4020                       <062>
4014 INPUT"{DOWN,2RIGHT}ALTES PROGRAMM LOE
     SCHEN??{2SPACE}(J/N)";A$:IF A$<>"J"TH
     EN RETURN                               <135>
4020 INPUT"{3DOWN,5RIGHT}FILENAME:";NA$      <185>
4030 OPEN 1,8,2,NA$+",S,R":GOSUB 5000        <181>
4035 PRINT"{DOWN,5RIGHT}";DS$:IF D1>19 THE
     N 4100                                  <008>
4040 INPUT#1,LL:MA=LL-1                      <062>
4050 FOR L=1 TO LL:INPUT#1,D:POKE AN+L,D:N
     EXT L                                   <250>
4100 CLOSE 1:CLOSE 15:IF D1>19 THEN 4020     <231>
4110 PRINT"{DOWN,5RIGHT}FILE EINGELESEN":S
     V=1:GOSUB 6000                          <239>
4120 RETURN                                  <182>
4200 :                                       <178>
4300 :                                       <022>
5000 REM *STATUS*                            <098>
5010 OPEN 15,8,15:INPUT#15,D1,DS$,D2,D3:RE
     TURN                                    <089>
5050 :                                       <007>
5100 PRINT"{CLR}":OPEN 1,8,0,"$":A=0         <042>
5110 GET#1,X$,X$                             <078>
5120 GET#1,X$,X$                             <089>
5130 GET#1,X$,X$                             <099>
5140 GET#1,X$:IF ST<>0 THEN 5220             <053>
5150 IF X$<>CHR$(34)THEN 5140                <141>
5160 GET#1,X$:IF X$<>CHR$(34)THEN PRINT X$
     ;:GOTO 5160                             <131>
5170 IF A=0 THEN A=1:PRINT:GOTO 5180         <035>
5175 A=0:PRINT TAB(20);                      <170>
5180 GET#1,X$:IF X$=CHR$(32)THEN 5180        <210>
5190 GET#1,X$:IF X$<>""THEN 5190             <086>
5200 GET T$:IF T$<>""THEN GOSUB 5240         <097>
5210 IF ST=0 THEN 5120                       <241>
5220 PRINT                                   <017>
5230 CLOSE 1:GOTO 6000                       <220>
5240 GET T$:IF T$=""THEN 5240                <074>
5250 RETURN                                  <036>
5400 :                                       <103>
5500 REM *DISK*                              <159>
5510 PRINT"{CLR,5DOWN}1 ← SAVEN":PRINT"{DO
     WN}2 ← EINLESEN":PRINT"{DOWN}3 ← INHA
     LT"                                     <243>
5520 GET T$:IF T$=""THEN 5520                <100>
5525 WA=VAL(T$):IF WA<1 OR WA>3 THEN 5520    <098>
5530 ON WA GOTO 3000,4000,5100               <006>
5600 :                                       <047>
5700 :                                       <148>
6000 REM *WARTEN*                            <059>
6005 :                                       <198>
6010 PRINT "{2DOWN,13RIGHT}<< TASTE >>"      <126>
6020 GET T$:IF T$=""THEN 6020                <086>
6030 RETURN                                  <051>
7000 :                                       <173>
8000 :                                       <153>
9000 REM *TEST*                              <110>
9005 :                                       <138>
9010 PRINT"{CLR,3DOWN,4RIGHT}PERIPHERIETES
     T":PRINT"{3DOWN,2RIGHT,RVSON}M{RVOFF}
     ENU{2DOWN}"                             <024>
9015 FOR L=1 TO 8:BI(L)=0:NEXT L:D=0         <146>
9020 PRINT" 1{2SPACE}2{2SPACE}3{2SPACE}4{2
     SPACE}5{2SPACE}6{2SPACE}7{2SPACE}8":P
     RINT                                    <179>
9025 GOTO 9055                               <192>
9030 GET T$:IF T$=""THEN 9030                <040>
9035 IF T$="M"THEN RETURN                    <233>
9040 LB=VAL(T$):IF LB<1 OR LB>8 THEN 9030    <018>
9045 LB=HI(LB)                               <040>
9050 IF BI(LB)=1 THEN BI(LB)=0:GOTO 9054     <221>
9051 IF BI(LB)=0 THEN BI(LB)=1               <073>
9054 UF=1:GOSUB 290                          <097>
9055 PRINT"{UP}";:FOR L=8 TO 1 STEP-1:PRIN
     T BI(L);:NEXT L:PRINT                   <135>
9060 POKE PO,D:GOTO 9030                     <188>
9100 :                                       <233>
9200 :                                       <077>
10000 REM *MENU*                             <079>
10005 :                                      <118>
10010 PRINT"{CLR,3DOWN,9RIGHT}P R O.S T{2S
      PACE}64.1{2DOWN}"                      <154>
10030 PRINT"{DOWN}1 ← EINGABE":PRINT"{DOWN
      }2 ← EDITIEREN"                        <016>
10040 PRINT"{DOWN}3 ← DISKETTE":PRINT"{DOW
      N}4 ← LOESCHEN":PRINT"{DOWN}5 ← STAR
      TEN"                                   <037>
10050 PRINT"{DOWN}6 ← PERIPHERIETEST":PRIN
      T"{DOWN}7 ← ENDE"                      <243>
10120 GET T$:IF T$=""THEN 10120              <150>
10130 WA=VAL(T$):IF WA<1 OR WA>7 THEN 1012
      0                                      <157>
10140 ON WA GOSUB 100,500,5500,80,2000,900
      0,11000                                <086>
10150 GOTO 10000                             <071>
10200 :                                      <057>
10300 :                                      <158>
11000 REM *ENDE*                             <033>
11005 :                                      <097>
11010 INPUT"{CLR,5DOWN,5RIGHT}ALLES GESICH
      ERT??? (J/N)";A$:IF A$<>"J"THEN RETU
      RN                                     <068>
11020 POKE PO,0:POKE 56579,0                 <006>
11030 :                                      <123>
11040 END                                    <171>
11050 :                                      <143>
11055 :                                      <148>
11060 :                                      <153>
12025 REM DIRECTORY (5100-5400) FREI         <158>
12030 REM NACH COMMODORE VIC 1541 SDFD       <072>
12035 REM USER'S MANUAL S.47                 <036>
12040 REM USERPORTDISPLAY VON JAN            <044>
12045 REM LEGENHAUSEN, 64'ER 8/84            <033>
12050 REM           VIELEN DANK!!!           <020>
12055 :                                      <128>
12060 REM ZEILEN 0-5 UND 12025 FF.           <206>
12065 REM WEGLASSEN!! GILT AUCH FUER         <077>
12070 :                                      <143>
12075 REM ↑ SOLCHE ↑                         <099>
```

Listing zu Pro.St (Schluß)

# Durch POKEs zum Erfolg — Die Spiele-Trickkiste (2)

**Wieder einmal greifen wir tief in unsere Trickkiste und ermöglichen damit auch dem nicht so ganz versierten Spielefan einen Ausflug in höhere oder sogar höchste Spiel-Level.**

Stellen Sie sich vor, Sie hätten bei Ghostbusters 1 000 000 Dollar (!) Startkapital, könnten bei H.E.R.O. Ihre Feinde verschwinden lassen oder bei Pitfall endlich alle Bilder zu Gesicht bekommen. Zauberei? Nein, ein paar POKEs und andere Tricks reichen völlig aus.

Allerdings erheben die folgenden POKEs keinen Anspruch auf Funktionsfähigkeit bei jeder Version der angegebenen Spiele. Insbesondere kommt es natürlich dann zu Problemen, wenn das Programm mit einem Autostart versehen ist und man keinen Reset-Schalter zur Verfügung hat.

Auch die Freunde von Ghostbusters kommen diesmal zu ihrem Recht: Zwar haben wir für dieses Spiel noch keinen POKE zur Hand, aber einige Tips, wie man das reichlich magere Startkapital kräftig erhöhen kann. (ev)

## POKEs für den C 64

| | | | |
|---|---|---|---|
| Alligata Blagger | 1. | POKE | 53264,126 |
| | 2. | POKE | 3574,44 |
| Bagitman | 1. | POKE | 19013,189 |
| | 2. | POKE | 22236,255 |
| Bat Attack | | POKE | 11061,234 |
| Battle through Time | | POKE | 22045,255 |
| Battle Zone | | POKE | 8909,100 |
| Black Hawk | | POKE | 8290,255 |
| Bruce Lee | 1. | POKE | 5686,128 |
| | 2. | POKE | 5672,128 |
| Cavelon | | POKE | 23789,255 |
| China Miner | 1. | POKE | 34623,234 |
| | 2. | POKE | 34624,234 |
| | 3. | POKE | 34625,234 |
| Crisis Mountain | | POKE | 2665,238 |
| Dare Devil Dennis | 1. | POKE | 29173,255 |
| | 2. | POKE | 17958,(Tempo) |
| Dig Dug | | POKE | 10473,255 |
| Dinkey Doo | 1. | POKE | 12296,165 |
| | 2. | POKE | 11989,18 |
| Donkey Kong | | POKE | 12118,234 |
| Dragon Hawk | | POKE | 3477,255 |
| Evolution | | POKE | 6947,255 |
| Falcon Patrol | 1. | POKE | 16764,234 |
| | 2. | POKE | 16765,234 |
| Flak | | POKE | 4798,36 |
| Fort Apocalypse | 1. | POKE | 36364,234 |
| | 2. | POKE | 36365,234 |
| | 3. | POKE | 36366,234 |
| | 4. | POKE | 36367,234 |
| | 5. | POKE | 36339,153 |
| | 6. | POKE | 36339,153 |
| Frantic Freddy | 1. | POKE | 34535,24 |
| | 2. | POKE | 31887,255 |
| Galaxy | | POKE | 3369,230 |
| Ghouls | | POKE | 8367,255 |
| Hard Hat Mack | 1. | POKE | 16877,173 |
| | 2. | POKE | 8472,100 |
| Herby | | POKE | 7191,255 |
| H.E.R.O. | | POKE | 14652,25 |
| High Noon | | POKE | 18033,255 |
| Hunchback | 1. | POKE | 5704,50 |
| | 2. | POKE | 9521,44 |
| Jet Set Willy | | POKE | 11345,33 |
| Jumpman Junior | 1. | POKE | 9450,173 |
| | 2. | POKE | 9450,44 |
| Jungle Hunt | | POKE | 2242,165 |
| Kickman | | POKE | 7424,230 |
| Kid Grid | | POKE | 10020,234 |
| Lady Tut | | POKE | 2392,50 |
| Loderunner | | POKE | 7892,255 |
| Miner 2049'er | 1. | POKE | 9450,173 |
| | 2. | POKE | 2652,165 |
| | 3. | POKE | 2471,255 |
| Mr. Robot | | POKE | 11518,255 |
| Omega Race | | POKE | 6300,230 |
| Pakakuda | | POKE | 7015,234 |
| Pitfall | | POKE | 5393,255 |
| Punchy | | POKE | 15458,50 |
| Q-Bert | | POKE | 4446,173 |
| Quest for Tires | 1. | POKE | 7341,99 |
| | 2. | POKE | 11485,125 |
| | 3. | POKE | 14864,0 |
| Radar Rar Race | | POKE | 7194,234 |
| Revenge of the Mutant Camels | | POKE | 35518,250 |
| Robin Rescue | | POKE | 6144,234 |
| Roundabout | | POKE | 12843,234 |
| Shamus | 1. | POKE | 27185,169 |
| | 2. | POKE | 27185,165 |
| | 3. | POKE | 27185,141 |
| | 4. | POKE | 27185,54 |
| | 5. | POKE | 27185,189 |
| Shamus Case II | 1. | POKE | 15475,238 |
| | 2. | POKE | 3888,128 |
| | 3. | POKE | 15476,176 |
| Sheep in Space | | POKE | 35039,44 |
| Snokie | 1. | POKE | 30116,173 |
| | 2. | POKE | 33242,200 |
| | 3. | POKE | 33242,255 |
| Squish 'em | | POKE | 2562,100 |
| Super Scramble | | POKE | 4691,(Tempo) |
| Survivor | | POKE | 19563,255 |
| Wheelin' Wallie | 1. | POKE | 27427,173 |
| | 2. | POKE | 27916,173 |
| Zeppelin | 1. | POKE | 10081,100 |
| | 2. | POKE | 14337,100 |

(Martin Glänzer/Frank Hoffmann/Christian Veber)

## POKEs für den VC 20

| | | |
|---|---|---|
| Buck Rogers | POKE | 8825,36 |
| Choplifter | POKE | 8011,173 |
| Crazy Kong | POKE | 30624,173 |
| Moonpatrol | POKE | 8300,5 |
| Scramble | POKE | 4691,(Tempo) |
| Shamus | POKE | 18486,169 |

(Thomas Thissen)

## Mehr Geld bei Ghostbusters

| Name | Kontonummer | Betrag |
|---|---|---|
| NNN | 21314100 | 23000 |
| Peter | 31664300 | 29900 |
| Tuerk,the | 06660401 | 48600 |
| Butterfly | 04664701 | 60000 |
| NNN | 20406201 | 70000 |
| Poke | 22014,9 | 90000 |
| S,S | 1984 | 100000 |
| P,M | 1984 | 100000 |
| 12345 | 25393120 | 415500 |
| Herbie | 05250624 | 500000 |
| Butterfly | 50542224 | 512800 |

Die Super-Nummer jedoch stammt von Mathias Höschen: $ 1 Million (!) erhält man bei Ghostbusters, wenn man statt eines Namens nur die RETURN-Taste drückt und als Kontonummer »458« eingibt. Als Anzeige erscheint jetzt nur »000000«, aber nach dem ersten Einkauf merkt man, daß man damit eine Million zur Verfügung hat.

(Hermann Gorges/Mathias Höschen/Christoph Peter/ Peter Siegert/Dirk Vreden)

## Directory ohne Programmverlust

Häufig möchte man sich das Directory einer Diskette ansehen ohne das gerade im Speicher befindliche Programm zu zerstören. Wenn man das DOS 5.1 nicht geladen hat, behilft man sich meist mit der zeitaufwendigen Zwischenspeicherung des Programms auf der Diskette. Es geht jedoch auch einfacher und schneller. Geben Sie einfach den folgenden Befehl ein:

POKE 44, PEEK(46) + 1

Damit wird der Basic-Anfang auf einen freien Speicherbereich gestellt. Sie können jetzt wie gewohnt mit »LOAD "$" ,8« das Directory laden und anschließend auflisten.

Mit POKE 44,8 sind Sie dann wieder im eigentlichen Programm.

(Heinzpeter Oelkers)

## Commodore-Joystick verbessert

Der Commodore-Joystick VIC-1311 benötigt eine relativ große Hebelbewegung, um die Kontakte zu schließen. Bei Spielen, die eine hohe Reaktionsgeschwindigkeit erfordern, ist diese Eigenschaft sehr ungünstig. Man kann jedoch recht einfach Abhilfe schaffen:

Man entfernt die vier Schrauben an der Unterseite des Gehäuses und hebt den oberen Teil des Joysticks mit der Platine vorsichtig ab. Nun wird die Platine an den Durchtrittsöffnungen der Schrauben mit je einer etwa 1 Millimeter dicken Unterlegscheibe verstärkt. Die vier Unterlegscheiben können mit einem Tropfen Alleskleber (Vorsicht, nicht die Kontakte verkleben !) gegen Verrutschen gesichert werden. Danach wird der Joystick wieder zusammengeschraubt. Wenn Sie alles richtig gemacht haben und insbesondere keine Teile übriggeblieben sind, dann werden die Kontakte des Joysticks nun bei erheblich kleineren Hebelbewegungen geschlossen.

(M. Kunze)

## Sequentielle Datei als Basic-Programm laden

Es sind eine Reihe von Anwendungen denkbar, bei denen aus einer sequentiellen Datei auf Diskette oder Kassette ein lauffähiges Basic-Programm erstellt werden soll (Datenfernübertragung, Umwandlung von Textfiles in Basic-Programme). Der folgende Einzeiler macht's für den VC 20 möglich:

OPEN 1, ⟨Gerät⟩, ⟨Sekundäradresse⟩, "⟨Name⟩" : POKE 812, 238 : POKE 781, 1 : SYS 65478

Dieses Miniprogramm öffnet das File Nummer 1 als Eingabefile (anstelle der Tastatur). Außerdem wird der CLALL-Vektor des Betriebssystems auf ein »RTS« gesetzt, so daß beim Einlesen von Programmzeilen keine Files geschlossen werden.

Deshalb werden nach Eingabe der obigen Befehlszeile von der ausgewählten Datei so lange Zeilen eingelesen und anschließend sofort im Direktmodus ausgeführt, bis die Betriebssystem-Routine CLRCHN aufgerufen wird (zum Beispiel durch einen Syntaxfehler in den gelesenen Zeilen oder durch GET #1, A$).

Um in den normalen Eingabemodus zurückzukehren, muß nur »POKE 812, 239 : CLR« eingegeben werden.

Zum Ausprobieren: Laden Sie ein beliebiges Basic-Programm und geben Sie danach im Direktmodus ein:

OPEN 1,8,3,"LISTING,S,W" : CMD 1 : LIST : PRINT#1 : CLOSE 1

## Tips zum C 64

Mit »POKE 808,225« wird die STOP-Taste ausgeschaltet und das Programm kann nicht mehr angehalten werden. »POKE 808,237« schaltet STOP wieder ein.

Eine Repeat-Funktion für alle Tasten wird mit »POKE 650,128« aktiviert. »POKE 650,0« schaltet wieder auf den Normalzustand zurück (Repeat-Funktion nur für die Leertaste und die Cursortasten sowie bei DEL und INST). Mit »POKE 650, 64« wird auch die Wiederholfunktion für diese Tasten abgeschaltet.

»POKE 657, 128« verhindert die Umschaltung zwischen Text- und Grafikmodus mit der Kombination von Shift- und Commodore-Taste. »POKE 657,0« hebt dies wieder auf.

Die Cursorfarbe kann mit »POKE 646,x« festgelegt werden, wobei für »x« der entsprechende Farbcode einzusetzen ist.

»PRINT PEEK(186) ergibt die Gerätenummer des zuletzt angesprochenen Peripheriegerätes.

(Thomas Lopatic)

## Zwei Tips für den C 64

Die Speicherstellen 57 und 58 enthalten die Zeilennummer der aktuellen Basic-Zeile. Die Abfrage geschieht mit »PRINT PEEK(57) + 256 * PEEK(58)«.

Mit »PRINT PEEK(1)« kann abgefragt werden, ob eine Taste an der Datasette gedrückt ist. Es gibt drei mögliche Werte:

7: Taste gedrückt,
55: keine Taste gedrückt,
39: Taste gedrückt, aber Programmlauf unterbrochen.

Diese Abfragen sind für die benutzerfreundliche Programmierung von Kassettenoperationen recht nützlich.

(Wolfgang Meyer)

## VC 20-Grundversion simuliert

Mit der folgenden kurzen Routine lassen sich die meisten Grundversions- oder +3 KByte-Programme auch mit einer 8 KByte-Erweiterung laden und ausführen:

POKE 648, 30 : SYS 64821
POKE 4096, 0 : POKE 44, 16 : NEW

Danach wird das Programm ganz normal geladen und gestartet. Da der Bildschirmspeicher durch die kleine Routine an der gleichen Stelle wie in der Grundversion liegt, sind die meisten Grundversions-Programme ohne weitere Änderungen direkt lauffähig.

## VC 20-Tips

POKE 792, 34 : POKE 793, 3 — Nach Drücken der RESTORE-Taste führt der Computer einen Kaltstart aus.

SYS 65511 — Dieser Befehl schließt alle Dateien

POKE 198,0:WAIT 198, 1 — Der Computer wartet, bis eine Taste gedrückt wird.

SYS 64802 — Es wird ein vollständiger Kaltstart ausgeführt

SYS 64818 — Kaltstart des VC 20, ist schneller als SYS 64802

POKE 818,34 : POKE 819,253 : POKE 37150,2 — Damit wird das Abspeichern von Programmen verhindert (Kopierschutz). Nach der Eingabe von »SAVE« führt der Computer einen Kaltstart durch.

(Thomas Hahn)


64ER ONLINE


64ER ONLINE


64ER ONLINE

## Fehlerteufelchen




**Änderungsvorschlag zur Turtle-Grafik, Ausgabe 3/85, Seite 90**

Der POKE-Befehl lautet korrekt: POKE 44,18: POKE 18 * 256,0: NEW. In Zeile 960 ist das Komma nach der 3 wegzulassen, so daß 32 entsteht. In Zeile 1040 ist nach dem 6. Datum nochmal eine 29 einzufügen.

**Erst ein IEC-Bus öffnet Tür und Tor, Ausgabe 3/85, Seite 24**

Das IEEE 488-Interface von Luda erlaubt, im Gegensatz zu der Angabe in der Tabelle auf Seite 26, die Verwendung einer Datasette. Was in der Kernalversion wegfällt, sind die Routinen zur seriellen Datenübertragung. Sie werden erst durch Umschalten auf den Standardkernal wieder möglich. Das Luda-Interface überträgt auch das EOI-Signal.

**Mit 80 fängt das Leben an, Ausgabe 4/85, Seite 19**

Die 80-Zeichenkarte XL-80 wird nicht direkt von General Automation vertrieben, sondern nur über Händler. Der Preis ist also von Fall zu Fall zu erfragen, er liegt über 250 Mark.

**Sternenhimmel, Ausgabe 5/85, Seite 58**

In den Zeilen 50140, 50340, 50360, 50400, 50440, 50470, und 50510 muß zwischen den Anführungszeichen jeweils ein Cursor up durchgeführt werden. In Zeile 50300 zwischen den Anführungsstrichen ein CLR (Bildschirm löschen) einfügen. In Zeile 50210 muß der Variablen pi der Wert $\pi$ zugewiesen werden, also pi = $\pi$.

**Trickfilm mit dem C 64, Ausgabe 5/85, Seite 64**

Im Listing »Trick.Obj« auf Seite 65 ist in Zeile C1D8 das dritte Datum etwas schlecht zu lesen, es lautet: A5 und nicht A3!

## Sonderheft Abenteuerspiele

**Quasimodo, Seite 76**

Wenn die Fehlermeldung »Next without for« in der Zeile 715 auftritt, so geben Sie vor dem Starten im Direktmodus POKE 44,64:POKE 16384,0 ein, und die Sache funktioniert. Die fehlende GOTO-Anweisung in Zeile 2100 ist ohne Bedeutung für den Ablauf des Programms.

**Zeittunnel, Seite 93**

Bitte ergänzen Sie die Zeile 4870 auf Seite 99 wie folgt: 4870 PRINT"(DOWN,RIGHT, RED) ZERSCHMETTERT. DU BIST TOT.":GOTO 4450.

**Floppy 1541 jetzt für PC-1500, Ausgabe 5/85, Seite 9**

Bei der angegebenen Telefonnummer fehlte leider eine Null. Die korrekte Anschrift lautet: Ing. Rudolf W. Fankhauser, Postfach 1115, D-7893 Jestetten. Das ist die Postanschrift für Deutschland. Die Anschrift in der Schweiz: SRS, Micro Computer Products, Ing. Rudolf W. Fankhauser, Postfach 1207, CH-8213 Neunkirch/SH, Tel. (004153) 62593.

0041 ist die internationale Vorwahl für die Schweiz!

**Leserservice, Ausgabe 5/85, Seite 165**

Obwohl nicht durch das Diskettensymbol explizit gekennzeichnet, sind folgende Listings auf der Leserservice-Diskette für die Ausgabe 5 enthalten: VIC — das intelligente Programm, Seite 175; 6510, Seite 71; Mini-Grafik für den VC 20, Seite 69; Sternenhimmel, Seite 58.



# COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64er« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **Juli-Ausgabe** (erscheint am 14. Juni 85): Schicken Sie Ihren Anzeigentext bis zum 20. Mai 85 (Eingangsdatum beim Verlag) an »64er«. Später eingehende Aufträge werden in der **August-Ausgabe** (erscheint am 19. Juli 85) veröffentlicht.

**Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes.**
**Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen.**
Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64er« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 11,— je Zeile Text veröffentlicht.

## Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen




64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE



64ER ONLINE


64ER ONLINE


64ER ONLINE




(unverbindliche ...



jetzt jeden Monat.

3.000-Mark-Chance und vieles mehr.


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE

Private Kleinanzeigen

Gewerbliche Kleinanzeigen




64ER ONLINE

## Gewerbliche Kleinanzeigen




64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE

# Bücher, die zum C 64 gehören

**Wer seinen C 64 wirklich kennenlernen möchte, kommt um den Kauf zusätzlicher Literatur nicht herum. Wir haben für Sie eine Hitliste aus dem unüberschaubar groß gewordenen Angebot an begleitender Literatur zusammengestellt. Mit diesem Sortiment ist jeder Basic- oder Assemblerprogrammierer, aber auch der Hardware-Bastler bestens ausgestattet.**

## Das eigentliche Handbuch

»Alles über den Commodore 64« aus der Commodore-Sachbuchreihe (Band 1) heißt das Standardwerk zum C 64. Wie schön wäre es, wenn dieses gute Buch mit jedem C 64 anstelle des mageren Handbuches ausgeliefert würde. Erstmals auf der Hannover-Messe 1984 angeboten, erfüllt dieses Programmierhandbuch einen lange gehegten Wunsch vieler Programmierer: Der englischsprachige »Programmers Reverence Guide« wurde ins Deutsche übersetzt.

Das fast 500 Seiten starke Programmierhandbuch ist im wesentlichen in drei Abschnitte gegliedert: Die Programmierung in Basic, in Assembler und die Beschreibung der Hardware. In allen drei Abschnitten wird auf die besonderen Baugruppen (Grafik, Musik, Ein-/Ausgabe) des C 64 und deren Programmierung eingegangen. Dem Konzept folgend, kein Lehrbuch, sondern mehr ein Nachschlagewerk zu sein, sind die abgedruckten Beispiele relativ kurz gehalten. Gerade aus der Kürze der Darstellung leitet sich aber der wichtigste Vorteil dieses Buches ab: Es ist sehr leicht, Informationen zu den verschiedensten Problemen zu finden. Sogar bei Fragen zum generellen Konzept des C 64 hilft das Programmierhandbuch, denn am Ende des Buches ist ein kompletter Schaltplan des C 64 eingeheftet.

Es gibt kaum ein Buch zum C 64, das gleichermaßen umfassend informiert und trotzdem leicht verständlich geschrieben ist. Dabei bleibt der Wert des Buches auch mit steigenden Programmierfähigkeiten des Besitzers erhalten, denn viele Kapitel (Bausteinbeschreibung, Assemblerprogrammierung, Z80-Modul) sind erst für den fortgeschrittenen Programmierer wichtig. Das Programmierhandbuch »Alles über den Commodore 64« sollte neben jedem C 64 liegen, auch wenn derzeit noch 59 Mark zusätzlich investiert werden müssen.

Info: »Alles über den Commodore 64«, Commodore Sachbuchreihe Band 1, Bezug über Fachhandel, ISBN3-89133-000-6, Preis 59 Mark

## Das Handbuch für den Profi

Mit dem »Systemhandbuch zum Commodore 64 und VC 20« haben die Profis von Interface Age **das** Handbuch für den fortgeschrittenen Programmierer geschaffen. Die Autoren Ralph Babel, M. Krause und A. Dripke haben in dieses mehr als 300 Seiten starke Buch ihre ganzen Erfahrungen aus der jahrelangen Arbeit mit Commodore-Computern einfließen lassen. Die eindeutige Aussage dieses Buches liegt in der Einsicht, daß ein so hervorragender Computer wie der C 64 erst in Assembler richtig programmiert werden kann. Dazu bedarf es aber der Kenntnis jedes einzelnen Bausteins im C 64. Herabsteigend bis auf die Chip-internen Grundvoraussetzungen werden die Bedingungen der Programmierung des C 64 umfassend und leicht verständlich beschrieben. In den vielen ausführlichen Kapiteln räumen die Autoren mit Unklarheiten und Halbwahrheiten über den C 64 auf. Folgende Themen werden dabei behandelt:

— Der Basic-Interpreter (Interne Codierung, Token-Tabelle, Binärarithmetik, Darstellung und Ablage von Variablen, USR-Funktion, Steuercode-Tabelle
— Assembler (Assembler-Programm, Befehlsliste, Befehlserklärung, Adressierungsarten)
— Grafik und Farbe (Bildschirm- und Farbspeicher, Zeichengenerator, HiRes-Grafik, Sprites, Interrupt- und Grafikkontrolle, Screen Blanking, Smooth Scrolling, Registerübersicht, Farben, Sprite-Generator, Hardware des VIC)
— Funktionstasten
— Tonerzeugung (Tongeneratoren, Frequenzberechnung, ADSR-Funktion, Wellenformen, Tonerzeugung, A/D-Wandler, Registerübersicht, Hardware des SID)
— Ein-/Ausgabe (Serieller Bus, RS232, CIA-Chip, Portprogrammierung, Timer, Interrupt-Handling, Registerübersicht, Kontrollports, Joystick, Paddles, Lightpen, Datenspeicherung auf Kassette/Diskette)
— Adaption von CBM-Programmen
— Speicheraufteilung (Speicherübersicht, dokumentierte Memory-Map, CPU-Speicherverwaltung, Prozessorport, Banking, Pinbelegung der CPU)
— ROM-Listing (ausführlich kommentiert)

Besonders das ausführlich kommentierte ROM-Listing ist von unschätzbarem Wert, denn kaum ein Assembler-Programm, das nicht auf die Systemroutinen angewiesen ist. Leider sind alle Adressen in dezimaler Schreibweise aufgeführt. Besser wäre es gewesen, die vom Assembler her gewohnte hexadezimale Form zu verwenden. Auch wer das Betriebssystem an seine eigenen Bedürfnisse anpassen möchte, findet hier alle notwendigen Informationen. Dieses Buch trägt zu Recht den Namen »Systemhandbuch«. Kompromißlos richtet sich das Buch an den fortgeschrittenen Programmierer, trotzdem ist es leicht verständlich.

Info: Babel, M. Krause, A. Dripke »Das Interface Age Systemhandbuch zum Commodore 64 und VC 20«, Bezug durch Fachhandel, Preis: 74 Mark, ISBN 3-88986-001-X

## Der Klassiker

Obwohl das Haus Data Becker in den letzten Jahren den Computer-Literaturmarkt mit einer Vielzahl von Werken überschwemmt hat, gehört eines der ersten zu den besten. »64 Intern« war zu einer Zeit, als kaum jemand den Commodore kannte (beziehungsweise kaufte, weil er zu teuer war), das einzige Buch, das dem mit seinem Handbuch alleine gelassenen C 64-Besitzer von seinem Informationsdefizit befreien konnte.

Ähnlich einem Systemhandbuch aufgebaut, soll »64 Intern« durch gute Beschreibung der Hardware und ihrer Programmierung ein ständiger Begleiter beim Programmieren sein. Diesen Anspruch unterstreicht das kommentierte ROM-Listing (in hexadezimaler Form) am Ende des Buches. Die wichtigsten Kapitel dieses Buches sind:

— Hardware (CPU, Speicherbelegungspläne, User-Port, Expansion-Port)
— Tonprogrammierung (Der SID, Register, A/D-Wandler, Synthy 64)
— Grafik (Der VIC, Registerbeschreibung, Betriebsarten, Schnittstellen zum Prozessor)
— Ein/Ausgabebausteine (Register-Plan, Ports, Timer, die CIAs, Joystickprogrammierung)
— Der Basic-Interpreter (Erweiterung des Basic, Monitor-Programm, wichtige Kernal-Adressen, RS232, serieller Bus)
— Vergleich: VC 20 — CBM — C 64
— ROM-Listing

Ebenso wie das Interface Age Systemhandbuch ist »64 Intern« für den fortgeschrittenen Programmierer nahezu unverzichtbar. Obwohl sich einige Informationen beider Bücher überschneiden, wird gerade durch die unterschiedliche Erklärung in beiden Büchern mancher Sachverhalt erst richtig verständlich.

Info: Angershausen, Brückmann, Englisch, Gerits, 64 Intern, Data Becker, 4. Auflage 1984, 350 Seiten, Preis: 59 Mark

## Von Basic zu Assembler

»Das Commodore-Buch Band 4«, ein Leitfaden für Systemprogrammierer, ist die Fortsetzung der Buchreihe von Markt&Technik. Dem Basic entwachsen, strebt so mancher Programmierer nach Höherem. Dieses Buch wendet sich deshalb vor allem an den Personenkreis, der gerade an der Schwelle der Assemblerprogrammierung steht. Somit widmen sich auch die ersten Kapitel der Beschreibung des Prozessors und seiner Befehle. Sehr übersichtlich werden Register, Adressierungsarten und Darstellungsformen der Assemblerschreibweise erklärt. Damit der Programmierer die neuen Kenntnisse auch gleich in die Tat umsetzen kann, haben die Autoren H. L. Schneider und W. Eberl in ihrem Buch gleich zwei Assembler und einen Disassembler mitgeliefert. Die erste Assemblerversion ist in Basic geschrieben. Die zweite Version wurde in durch Maschinensprache unterstütztem Basic entwickelt. Selbstverständlich stehen zu jedem Programm ausführliche Beschreibungen und Anleitungen zur Verfügung. Weitere Kapitel widmen sich der Zusammenarbeit von Maschinenprogrammen mit Basic. Sehr gut gelungen ist dabei die Tabelle der Basic-ROM-Routinen. Sie helfen besonders bei der Erstellung eigener Maschinenprogramme, denn durch geschicktes Verwenden dieser Routinen wird viel Platz gespart. Der Leser lernt nebenbei rationell zu programmieren. Trotzdem ist der Leitfaden für Systemprogrammierer kein Assembler-Lehrbuch, sondern ein für den Assembler-Anfänger lehrreiches und für den Profi hilfreiches Buch, das in keiner Sammlung fehlen sollte. Der pädagogische Wert dieses Buches wird durch die für 58 Mark erhältliche Diskette mit allen Programmen unterstützt.

(Arnd Wängler)

Info: H. L. Schneider, W. Eberl, Das Commodore 64-Buch, Band 4, ISBN-Nr.: 3-922120-70-9, Markt & Technik Verlag Haar b. München. Preis: 38 Mark

# In die Geheimnisse der Floppy eingetaucht (Teil 7)

**Dieser Teil zeigt wie das DOS codiert und physikalisch auf Diskette schreibt. Haben Sie gewußt, daß ein Block auf Diskette länger als 256 Byte ist?**

Dieser Artikel bildet den Abschluß der festen Kursreihe. Ab jetzt soll der Floppy-Kurs in lockerer Reihenfolge fortgesetzt werden. Wir werden nun versuchen, auf die von Ihnen bevorzugten Themen einzugehen, um darüber dann in unregelmäßigen Abständen weitere Folgen zu bringen. Wir wollen Sie deshalb dazu anregen, uns Ihre Probleme und Sorgen mit der 1541 mitzuteilen.

An dieser Stelle soll aber noch ein wichtiger Sachverhalt besprochen werden, der bisher einfach unterschlagen wurde: die GCR-Codierung.

## Was ist eine GCR-Codierung?

Wenn Sie den Floppy-Kurs schon länger verfolgen sind Ihnen bestimmt schon einige Ungereimtheiten aufgefallen, was den Direktzugriff auf die Diskette betrifft. Auch in der letzten Folge über das Formatieren waren zum Beispiel im Listing von S-Format einige Sprungbefehle, die nicht erklärt wurden.

Erinnern Sie sich noch an den Artikel, der sich das erste mal mit dem Schreiben von Daten auf die Diskette beschäftigte? Dort wurden, unter anderem, die SYNC-Markierungen auf der Diskette besprochen, die dem Disk-Controller als Positionsanzeiger dienen.

Ich schrieb damals, daß sich diese SYNC-Markierungen bei der 1541 aus fünf $FF-Bytes zusammensetzen, die hintereinander auf Diskette geschrieben werden. Was ist aber, wenn ein Datenblock geschrieben werden soll, der nur aus $FF-Bytes besteht? Eigentlich müßten dann diese Bytes als SYNC-Markierung wirksam werden und den gesamten Schreib- und Lesebetrieb stören. Wie die Praxis zeigt, tritt dieser Fehler nicht auf. Auch bei mehreren Blöcken aus $FF-Bytes kommt es zu keinen Komplikationen. Bei der Konstruktion der Floppy hat man sich nämlich eine Codierung der Daten einfallen lassen, die eine Eindeutigkeit der Daten schafft. Die Codierung heißt GCR, was nichts anderes als eine Abkürzung der englischen Wörter »Group Code Recording« ist.

Es stellt sich jetzt natürlich die Frage, was bei der GCR-Codierung passiert, damit eine Verwechselung zwischen SYNC- und Datenbytes unmöglich wird. Zur Beantwortung dieser Frage muß ein wenig intensiver auf das Lesen und Schreiben der Floppy eingegangen werden.

## Was macht die GCR-Codierung

Das Lesen von Bytes durch den Lesekopfes steuert ein Timer des Disk-Controllers. Auf der Diskette selbst wird jedes »1«-Bit physikalisch durch einen Wechsel der Magnetisierungsrichtung dargestellt und ein »0«-Bit durch gleichbleibende Richtung der Magnetisierung. Bild 1 zeigt, was gemeint ist. Soll ein Byte von Diskette gelesen werden, so wartet der Disk-Controller einfach die Zeitspanne ab, die zum Lesen von acht Bits erforderlich ist. Innerhalb dieser Zeit liest der Schreib-/Lesekopf eine gewisse Folge von Magnetisierungswechseln und Nicht-Magnetisierungswechseln.

Dazu ein Beispiel: Auf der Diskette steht ein $55-Byte. $55 wird binär durch die Kombination %01010101 dargestellt. Der Tonkopf stellt also während der Lesezeit die folgenden Magnetisierungswechsel fest:

Magnetisierung wechselt nicht, wechselt, wechselt nicht, wechselt, wechselt nicht, wechselt, wechselt nicht, wechselt.

Das Erkennen eines Bits geschieht dabei völlig zeitgesteuert. Der Disk-Controller »weiß«, daß er zum Lesen eines Bits eine bestimmte Zeit warten muß. Danach gilt das Bit als gelesen, und es wird eine »1« oder eine »0« bereitgestellt, je nachdem, ob ein Magnetisierungswechsel stattgefunden hat oder nicht.

Praktisch könnte man das folgendermaßen beschreiben: Sie machen mit einem Freund eine Zeit von 10 Sekunden aus. Er hat dann die Aufgabe innerhalb dieser 10 Sekunden entweder zu pfeifen oder nicht. Danach warten Sie diese 10 Sekunden ab. Hat er während dieser Zeit gepfiffen, dann entspricht das einem Magnetisierungswechsel. Hat er innerhalb der 10 Sekunden nicht gepfiffen, bedeutet das ein »0«-Bit, also keinen Magnetisierungswechsel. Da eine Diskette im Laufwerk nicht absolut gleichmäßig gedreht werden kann, also Drehzahlschwankungen unterliegt, muß noch für eine Kompensation der mechanischen Fehler gesorgt werden. Dazu wird der Timer, der die abzuwartende Zeit für jedes Bit bestimmt, bei jedem Magnetisierungswechsel neu getriggert (gestellt). Ein »1«-Bit hat also neben seinem Informationsgehalt noch die wichtige Aufgabe, Laufwerksschwankungen auszugleichen, um Lesefehler zu verhindern. Aus diesem Grund darf es zum Beipiel nicht passieren, daß mehrere $00-Bytes hintereinander auf der Diskette stehen, da sonst zu lange keine Laufwerkskontrolle mehr stattfinden könnte.

Aber auch zuviele »1«-Bits sind nicht gestattet, da mehr als acht »1«-Bits ein SYNC-Signal auslösen.

Aus den genannten Gründen werden alle Daten, die auf die Diskette geschrieben werden, vorher GCR-codiert. Mit dieser Codierung wird ausgeschlossen, daß mehr als acht »1«-Bits und mehr als zwei »0«-Bits direkt hintereinander auf die Diskette geschrieben werden und so die Schreib- und Lese-Elektronik durcheinanderbringen.

Einzig und allein die SYNC-Markierungen (mehr als acht »1«-Bits) werden vom DOS (Disk Operating System, Controller) uncodiert auf die Diskette geschrieben.

## Es gibt zwei Schreibarten

Man kann also zwischen zwei Schreibarten auf Diskette unterscheiden:

1) Schreiben von Markierungen.
Hier werden fünf $FF-Bytes direkt hintereinander auf die Diskette geschrieben, um eine SYNC-Markierung zu bilden, die der Orientierung dient.

2) Schreiben von Daten.
In diesem Modus werden Byte-Inhalte codiert, um sich von den Markierungen zu unterscheiden.

Sehen Sie sich jetzt einmal Tabelle 1 an, die Umwandlungstabelle für die Konvertierung Binär nach GCR und umgekehrt.

Wie Sie unschwer erkennen können, handelt es sich beim GCR-Code um einen 5-Bit-Code. Jedes 4-Bit-Nibble, das Sie umwandeln, wird zu einem 5-Bit-GCR-Nibble. Ein Byte, das vorher aus 8 Bits bestand wird also durch die Codierung 10 Bits lang. Allgemein nimmt die Länge der codierten Daten, um den Faktor 5/4 zu. Deshalb ist die Handhabung der GCR-Bytes nicht ganz einfach. Wandeln Sie doch einmal zwei Bytes in den GCR-Code um. Als Ergebnis erhalten Sie »zweieinhalb« Bytes, die sicherlich schwer zu behandeln sind.

Bei der GCR-Codierung geht man aus diesem Grund einen ganz einfachen Weg, um keine Format-Probleme zu bekommen: Es werden jeweils immer 4 Bytes gleichzeitig umgewandelt. Als Ergebnis erhält man 5 vollständige Bytes, die sich ohne Probleme weiterverarbeitet werden können.

Lassen Sie mich das einmal an einem Beipiel erläutern:

Nehmen wir einmal an, wir hätten vier Bytes mit dem Wert $FF. Eine Kombination also, die nicht direkt auf die Diskette geschrieben werden darf.

Wir wandeln diese vier Hex-Bytes nun in die entsprechenden fünf GCR-Bytes um, indem wir in Tabelle 1 nachsehen, was die entsprechenden GCR-Äquivalente dieser Bytes sind. Wir kommen zu folgendem Ergebnis:

| HEX | BINÄR | GCR-Code |
|---|---|---|
| $FF | 1111 1111 | 10101 10101 |
| $FF | 1111 1111 | 10101 10101 |
| $FF | 1111 1111 | 10101 10101 |
| $FF | 1111 1111 | 10101 10101 |

Die binär dargestellten GCR-Werte müssen wir jetzt nur noch zu fünf Bytes zusammenfassen, um auf folgendes Ergebnis zu kommen:

1010 + 1101 = AD
**(1010 1+101** 01)
0110 + 1011 = 6B
0101 + 1010 = 5A
1101 + 0110 = D6
1011 + 0101 = B5

Vier $FF-Bytes werden also bei der GCR-Codierung in die fünf Bytes $AD, $6B, $5A, $D6 und $B5 umgewandelt, und Sie können sich jetzt leicht davon überzeugen, daß diese fünf Bytes für den Disk-Controller absolut ungefährlich und unkritisch

sind, und daß sie die vorgeschriebenen Normen (nicht mehr als zwei »0«-Bytes und nicht mehr als acht »1«-Bytes) erfüllen.

Um Ihnen die Umwandlung der Bytes zu erleichtern, habe ich diesem Kurs zwei Programmlistings beigefügt. Listing 1 enthält ein Programm, das Ihnen vier Hex-Bytes in fünf GCR-Bytes umwandelt. In Listing 2 sehen Sie ein Programm abgedruckt, das die GCR-Codierung wieder rückgängig macht. Hier werden fünf GCR-Bytes in vier Hex-Bytes zurückverwandelt, wobei Sie mit unerlaubten Bitkombinationen vorsichtig sein sollten. Kann ein Byte nicht zurückverwandelt werden, so haben Sie eine unerlaubte GCR-Bitkombination, die sich im Ergebnis dadurch äußert, daß entsprechende Nibbles fehlen. Sie erhalten dann unter Umständen nur »halbe« Bytes.

Die Floppy hält übrigens für diesen Fall eine Fehlermeldung bereit, einen »24, READ ERROR«.

Damit Sie auch in Maschinensprache in der Lage sind, Hex-GCR-Konvertierungen durchzuführen, ist noch ein weiteres Listing (Listing 3) beigefügt. Dieses enthält die Originalroutinen des DOS zur Umwandlung von Hex-Bytes in GCR-Bytes und umgekehrt.

**$F6D0**: Dieses Programm holt vier Hex-Bytes aus den Speicherstellen $52 bis $55 und wandelt diese Bytes in die fünf entsprechenden GCR-Werte um. Diese fünf Bytes werden anschließend im Puffer der Adresse $30/31 (L,H) mit dem Pufferzeiger in $34 abgelegt.

Pufferadresse und Pufferzeiger müssen dabei vor Aufruf dieser Routine übergeben werden.

**$F78F**: Diese Routine wandelt einen gesamten Puffer, dessen Adresse in $30/31 (L,H) stehen muß, in GCR-Werte um und speichert diese in den Ausweichpuffer sowie den ursprünglichen Puffer zurück. Der Pufferinhalt vergrößert sich durch diese Umwandlung von 256 auf 324 Bytes.

**$F7E6**: Diese Routine wandelt fünf GCR-Bytes aus einem Puffer (dessen Adresse in $30/31 (L,H) und dessen Pufferzeiger in $34 steht) wieder in vier Hex-Bytes zurück, wobei diese dann in der Zeropage von $52 bis $55 abgespeichert werden.

**$F8E0**: Diese Routine decodiert einen gesamten GCR-Pufferinhalt in die ursprüngliche Form und legt diese 256 Bytes dann im Puffer mit der Adresse $30/31 (L,H) ab. Die vorherigen 324 GCR-Bytes müssen im gleichen Puffer und im Ausweichpuffer ($01BB bis $01FF) stehen.

Die Anwendungen dieser Routinen sind äußerst vielfältig. So können Sie diese Programme zum Beispiel für einen Disk-Monitor verwenden, in dem man zwischen der Anzeige von GCR-Bytes und der Anzeige von normalen Hex-Bytes hin- und herschalten kann. Die einzigen Änderungen, die Sie dazu machen müssen, bestehen in der Umrechnung der Adressen für die Speicherbereiche im Computer und der Angabe neuer Parameter als Puffer- und Zeropagebereiche. Ihrer Phantasie, was die Möglichkeiten des Monitors angeht, sind außer dem Speicherplatz im Computer keine Grenzen gesetzt.

## Bis zu 365 Byte in einem Block

Durch die Verwendung der GCR-Codierung ergeben sich noch Konsequenzen. Wie sieht es beispielsweise in den Puffern der Floppy aus, wenn ein Puffer mit einem vollständigen Datenblock (also 256 Bytes) gefüllt wurde und dieser aufgezeichnet werden soll? Für dieses Problem hat der Controller einen speziellen Ausweichpuffer. Der Puffer hat eine Größe von 68 Bytes und befindet sich im Bereich von $01BB bis $01FF.

Wird nun ein Datenblock in Puffer 1 ($0400-$04FF) codiert, so werden die ersten 68 GCR-Bytes in den Ausweichpuffer übernommen. Die restlichen Bytes stehen in Puffer 1.

Aus den 256 Bytes an Information macht das DOS durch die Konvertierung also 324 Bytes, die einen gesamten Datenblock darstellen (inklusive Prüfsumme). Natürlich werden auch die Parameter im Datenblockheader (ID, Track, Sektor, Prüfsumme und Kennzeichen) vor dem Schreiben auf die Diskette in GCR-Bytes umgewandelt, wobei der Blockheader dann mit den zwei Lückenbytes auf eine Länge von zehn GCR-Bytes anwächst, da der Header aus ursprünglich acht Hex-Werten besteht.

Zusammenfassend besteht ein Sektor auf der Diskette aus den fünf Bytes der ersten SYNC-Markierung; danach folgen die zehn Bytes des Blockheaders. Vor der SYNC-Markierung des Datenblocks folgen jedoch noch neun $55-Bytes, die der GCR-Norm entsprechen und direkt auf die Diskette geschrieben werden. Sie dienen als Pufferlücke, in der dem Disk-Controller Zeit bleibt, zwischen Schreiben und Lesen umzuschalten.

Nach den fünf Bytes der SYNC-Markierung folgen die 324 Bytes des Datenblocks inklusive dessen Prüfsumme und anschließend noch die Lücke zwischen zwei Sektoren, die erfahrungsgemäß zwischen acht und zwölf Bytes lang ist. Wie Sie sehen hat also so ein Sektor auf der Diskette die stattliche Länge von 361 bis 365 Bytes.

Jetzt werden Ihnen bestimmt auch ein paar zweifelhafte JSR-Befehle in der letzten Folge des Floppy-Kurses klar: bei dem Formatiersystem in Ausgabe 5/1985, wird einmal ein Befehl JSR $FE30 und an anderer Stelle ein Befehl JSR $F78F ausgeführt. Diese Adressen sind die Einsprünge der Codier-Routinen.

Vielleicht kommt Ihnen auch noch einmal die Herstellung eines Killertracks in Erinnerung. Hier wird ein gesamter Track direkt mit $FF-Bytes vollgeschrieben und stellt so eine »Riesen-SYNC-Markierung« dar. Da eine solche Bitfolge jedoch unzulässig ist, kommt die Lese-und Schreibelektronik der Floppy völlig aus dem Konzept; der Controller »stürzt ab«.

Wenn Sie noch mehr über Ihre 1541, über schnelle Kopierprogramme und Kopierschutz-Methoden erfahren oder ein gut dokumentiertes DOS-Listing haben wollen, dann sollten Sie einmal in das M&T Floppy-Buch schauen.(Karsten Schramm/hm)

```
N/N      S/S  N/N  S/S      N/N  S/S           N/N      S/S
 *        *    *    *        *    *             *        *

1   0   0   1   1   1   0   1   1   0   0   1   0   1
```

* = Magnetisierungswechsel

**Bild 1. Die Aufzeichnung von Daten auf Diskette (schematisch).**

```
10 REM PROGRAMM ZUR KONVERTIERUNG              <242>
20 REM VON FUENF GCR-BYTES IN DIE              <003>
30 REM VIER ENTSPRECHENDEN                     <249>
40 REM HEX-AEQUIVALENTE                        <077>
50 REM                                         <193>
60 REM                                         <203>
70 REM                                         <213>
80 REM (W) 1985 BY KARSTEN SCHRAMM             <028>
90 REM                                         <233>
100 A$="0123456789ABCDEF":DIM G$(15):E$=""    <220>
110 G$(0)="01010"                              <066>
120 G$(1)="01011"                              <078>
130 G$(2)="10010"                              <088>
140 G$(3)="10011"                              <100>
150 G$(4)="01110"                              <111>
160 G$(5)="01111"                              <123>
170 G$(6)="10110"                              <133>
180 G$(7)="10111"                              <145>
190 G$(8)="01001"                              <154>
200 G$(9)="11001"                              <166>
210 G$(10)="11010"                             <216>
220 G$(11)="11011"                             <228>
230 G$(12)="01101"                             <238>
240 G$(13)="11101"                             <250>
250 G$(14)="11110"                             <005>
260 G$(15)="10101"                             <016>
270 PRINT"{CLR}GCR - HEX - KONVERTIERUNG":
    PRINT                                      <096>
280 PRINT:PRINT"GEBEN SIE JETZT 5 GCR-BYTE
    S EIN":PRINT                               <016>
290 INPUT"{2DOWN}";H$:GC$=""                   <147>
300 X$="":FOR X=1 TO LEN(H$)                   <185>
310 IF MID$(H$,X,1)<>" "THEN X$=X$+MID$(H$
    ,X,1)                                      <053>
320 NEXT                                       <195>
330 H$=X$                                      <229>
340 FOR X=1 TO 10                              <022>
350 X$=MID$(H$,X,1)                            <245>
360 XX=VAL(X$):IF XX=0 AND X$<>"0"THEN XX=
    ASC(X$)-55                                 <104>
370 FOR Y=0 TO 3                               <006>
380 YY=INT(XX/2↑(3-Y)):XX=XX-YY*2↑(3-Y)        <105>
390 IF YY THEN GC$=GC$+"1":GOTO 410            <240>
400 GC$=GC$+"0"                                <189>
410 NEXT Y,X                                   <250>
420 HC$="":FOR X=1 TO 8                        <028>
430 X$=MID$(GC$,X*5-4,5)                       <075>
440 FOR Y=0 TO 15                              <127>
450 IF X$<>G$(Y)THEN NEXT Y                    <197>
460 :                                          <007>
470 HC$=HC$+MID$(A$,Y+1,1)                     <206>
480 IF INT(X/2)=X/2 THEN HC$=HC$+" "           <055>
490 NEXT X                                     <197>
500 PRINT:PRINT:PRINT"HEX: ";HC$               <129>
```

**Listing 1. Umwandlung von Daten in GCR-Bytes**

```
10 REM PROGRAMM ZUR KONVERTIERUNG         <242>
20 REM VON VIER HEXBYTES IN DIE           <161>
30 REM FUENF ENTSPRECHENDEN               <055>
40 REM GCR-AEQUIVALENTE                   <068>
50 REM                                    <193>
60 REM                                    <203>
70 REM                                    <213>
80 REM (W) 1985 BY KARSTEN SCHRAMM        <028>
90 REM                                    <233>
100 A$="0123456789ABCDEF":DIM G$(15):E$="" <220>
110 G$(0)="01010"                         <066>
120 G$(1)="01011"                         <078>
130 G$(2)="10010"                         <088>
140 G$(3)="10011"                         <100>
150 G$(4)="01110"                         <111>
160 G$(5)="01111"                         <123>
170 G$(6)="10110"                         <133>
180 G$(7)="10111"                         <145>
190 G$(8)="01001"                         <154>
200 G$(9)="11001"                         <166>
210 G$(10)="11010"                        <216>
220 G$(11)="11011"                        <228>
230 G$(12)="01101"                        <238>
240 G$(13)="11101"                        <250>
250 G$(14)="11110"                        <005>
260 G$(15)="10101"                        <016>
270 PRINT"{CLR}HEX - GCR - KONVERTIERUNG":
    PRINT                                 <096>
280 PRINT:PRINT"GEBEN SIE JETZT 4 HEXBYTES
     EIN":PRINT                           <235>
290 PRINT"Z.B. ED 34 27 58":INPUT"{2DOWN}"
    ;H$:GC$=""                            <104>
300 GOSUB 470:FOR X=1 TO 4                <035>
310 H1$=MID$(H$,X*2-1,1):H2$=MID$(H$,X*2,1
    )                                     <120>
320 H1=VAL(H1$):H2=VAL(H2$)               <057>
330 IF H1=0 AND H1$<>"0"THEN H1=ASC(H1$)-5
    5                                     <240>
340 IF H2=0 AND H2$<>"0"THEN H2=ASC(H2$)-5
    5                                     <254>
350 GC$=GC$+G$(H1)+G$(H2)                 <044>
360 NEXT X                                <067>
370 FOR X=1 TO 10                         <052>
380 B=0:B$=MID$(GC$,X*4-3,4)              <094>
390 FOR Y=0 TO 3                          <026>
400 IF MID$(B$,Y+1,1)="1"THEN B=B+2↑(3-Y) <112>
410 NEXT Y                                <118>
420 E$=E$+MID$(A$,B+1,1)                  <249>
430 IF X/2=INT(X/2)THEN E$=E$+" "         <121>
440 NEXT X                                <147>
450 PRINT:PRINT:PRINT"GCR:  ";E$          <000>
460 END                                   <077>
470 X$="":FOR X=1 TO LEN(H$)              <099>
480 IF MID$(H$,X,1)<>" "THEN X$=X$+MID$(H$
    ,X,1)                                 <223>
490 NEXT                                  <109>
500 H$=X$:RETURN                          <087>
```

**Listing 2. Umwandlung von GCR- in Daten-Bytes ▲**

| Hexa-dezimal | Binär | GCR |
|---|---|---|
| $0 | 0000 | 01010 |
| $1 | 0001 | 01011 |
| $2 | 0010 | 10010 |
| $3 | 0011 | 10011 |
| $4 | 0100 | 01110 |
| $5 | 0101 | 01111 |
| $6 | 0110 | 10110 |
| $7 | 0111 | 10111 |
| $8 | 1000 | 01001 |
| $9 | 1001 | 11001 |
| $A | 1010 | 11010 |
| $B | 1011 | 11011 |
| $C | 1100 | 01101 |
| $D | 1101 | 11101 |
| $E | 1110 | 11110 |
| $F | 1111 | 10101 |

**Tabelle 1. Umrechnungstabelle für Binär-GCR-Umwandlung**

**Listing 3. Die DOS-Routinen zur GCR-Codierung ▼**

```
Umwandlung von 4 Daten-Bytes
in 5 GCR-Bytes
f6d0 a9 00      lda  #$00
f6d2 85 57      sta  $57
f6d4 85 5a      sta  $5a
;    Pufferzeiger holen
f6d6 a4 34      ldy  $34
;    erstes Byte holen
f6d8 a5 52      lda  $52
f6da 29 f0      and  #$f0
f6dc 4a         lsr
f6dd 4a         lsr
f6de 4a         lsr
f6df 4a         lsr
f6e0 aa         tax
;    und anhand der Tabelle
;    umwandeln
f6e1 bd 7f f7   lda  $f77f,x
f6e4 0a         asl
f6e5 0a         asl
f6e6 0a         asl
f6e7 85 56      sta  $56
f6e9 a5 52      lda  $52
f6eb 29 0f      and  #$0f
f6ed aa         tax
f6ee bd 7f f7   lda  $f77f,x
f6f1 6a         ror
f6f2 66 57      ror  $57
f6f4 6a         ror
f6f5 66 57      ror  $57
f6f7 29 07      and  #$07
f6f9 05 56      ora  $56
;    Byte in Puffer schreiben
f6fb 91 30      sta  ($30),y
f6fd c8         iny
;    zweites Byte holen
f6fe a5 53      lda  $53
f700 29 f0      and  #$f0
f702 4a         lsr
f703 4a         lsr
f704 4a         lsr
f705 4a         lsr
f706 aa         tax
f707 bd 7f f7   lda  $f77f,x
;    und codieren
f70a 0a         asl
f70b 05 57      ora  $57
f70d 85 57      sta  $57
f70f a5 53      lda  $53
f711 29 0f      and  #$0f
f713 aa         tax
f714 bd 7f f7   lda  $f77f,x
f717 2a         rol
f718 2a         rol
f719 2a         rol
f71a 2a         rol
f71b 85 58      sta  $58
f71d 2a         rol
f71e 29 01      and  #$01
f720 05 57      ora  $57
;    in Puffer schreiben
f722 91 30      sta  ($30),y
f724 c8         iny
;    drittes Byte holen
f725 a5 54      lda  $54
f727 29 f0      and  #$f0
f729 4a         lsr
f72a 4a         lsr
f72b 4a         lsr
f72c 4a         lsr
f72d aa         tax
;    codieren und in
f72e bd 7f f7   lda  $f77f,x
f731 18         clc
f732 6a         ror
f733 05 58      ora  $58
;    Puffer schreiben
f735 91 30      sta  ($30),y
f737 c8         iny
f738 6a         ror
f739 29 80      and  #$80
f73b 85 59      sta  $59
f73d a5 54      lda  $54
f73f 29 0f      and  #$0f
f741 aa         tax
f742 bd 7f f7   lda  $f77f,x
f745 0a         asl
f746 0a         asl
f747 29 7c      and  #$7c
f749 05 59      ora  $59
f74b 85 59      sta  $59
;    viertes Byte holen
f74d a5 55      lda  $55
f74f 29 f0      and  #$f0
f751 4a         lsr
f752 4a         lsr
f753 4a         lsr
f754 4a         lsr
f755 aa         tax
;    codieren und in
f756 bd 7f f7   lda  $f77f,x
f759 6a         ror
f75a 66 5a      ror  $5a
f75c 6a         ror
f75d 66 5a      ror  $5a
f75f 6a         ror
f760 66 5a      ror  $5a
f762 29 03      and  #$03
f764 05 59      ora  $59
;    Puffer schreiben
f766 91 30      sta  ($30),y
f768 c8         iny
f769 d0 04      bne  $f76f
f76b a5 2f      lda  $2f
f76d 85 31      sta  $31
f76f a5 55      lda  $55
f771 29 0f      and  #$0f
f773 aa         tax
f774 bd 7f f7   lda  $f77f,x
f777 05 5a      ora  $5a
;    Überlaufbyte ebenfalls
;    in Puffer schreiben
f779 91 30      sta  ($30),y
f77b c8         iny
;    Pufferzeiger merken
f77c 84 34      sty  $34
f77e 60         rts
;    Tabelle für die
;    Umwandlung
;    von HEX nach GCR-
;    Werten
f77f 0a 0b 12 13 0e 0f 16 17
f787 09 19 1a 1b 0d 1d 1e 15

;    Pufferdaten codieren
;    Pufferadresse Lo
f78f a9 00      lda  #$00
f791 85 30      sta  $30
f793 85 2e      sta  $2e
;    Pufferzeiger 2 setzen
f795 85 36      sta  $36
;    Pufferzeiger für
f797 a9 bb      lda  #$bb
;    Ausweichpuffer setzen
f799 85 34      sta  $34
f79b 85 50      sta  $50
;    Pufferadresse Hi
f79d a5 31      lda  $31
;    merken
f79f 85 2f      sta  $2f
;    Pufferadresse auf
f7a1 a9 01      lda  #$01
;    Ausweichpuffer
f7a3 85 31      sta  $31
;    Datenblockkennz. $07
;    für Kodierung
;    vorbereiten
f7a5 a5 47      lda  $47
f7a7 85 52      sta  $52
f7a9 a4 36      ldy  $36
;    Bytes aus normalem
;    Puffer
f7ab b1 2e      lda  ($2e),y
;    in Speicher für
;    Codierung
f7ad 85 53      sta  $53
f7af c8         iny
f7b0 b1 2e      lda  ($2e),y
f7b2 85 54      sta  $54
f7b4 c8         iny
f7b5 b1 2e      lda  ($2e),y
f7b7 85 55      sta  $55
f7b9 c8         iny
;    Pufferzeiger retten
f7ba 84 36      sty  $36
;    Bytes kodieren und in
;    Ausweichpuffer
;    schreiben
f7bc 20 d0 f6   jsr  $f6d0
f7bf a4 36      ldy  $36
;    weitere Bytes aus
;    Puffer in
f7c1 b1 2e      lda  ($2e),y
;    Speicher für Codierung

f7c3 85 52      sta  $52
f7c5 c8         iny
f7c6 f0 11      beq  $f7d9
f7c8 b1 2e      lda  ($2e),y
f7ca 85 53      sta  $53
f7cc c8         iny
f7cd b1 2e      lda  ($2e),y
```

```
f7cf  85 54      sta  $54
f7d1  c8         iny
f7d2  b1 2e      lda  ($2e),y
f7d4  85 55      sta  $55
f7d6  c8         iny
;     und ggf. codieren
f7d7  d0 e1      bne  $f7ba
;     Prüfsumme Datenblock
f7d9  a5 3a      lda  $3a
;     ebenfalls codieren und
;     in Puffer schreiben
f7db  85 53      sta  $53
f7dd  a9 00      lda  #$00
f7df  85 54      sta  $54
;     Leerbytes zum Auffüllen
f7e1  85 55      sta  $55
;     ebenfalls codieren
f7e3  4c d0 f6   jmp  $f6d0
Umwandlung von 5 GCR-Bytes
in 4 Daten-Bytes
;     Pufferzeiger holen
f7e6  a4 34      ldy  $34
;     Byte aus Puffer holen
f7e8  b1 30      lda  ($30),y
f7ea  29 f8      and  #$f8
f7ec  4a         lsr
f7ed  4a         lsr
f7ee  4a         lsr
f7ef  85 56      sta  $56
;     und decodieren;
;     zwischen-
;     speichern für späteres
f7f1  b1 30      lda  ($30),y
f7f3  29 07      and  #$07
f7f5  0a         asl
f7f6  0a         asl
;     zusammenschieben
f7f7  85 57      sta  $57
f7f9  c8         iny
f7fa  d0 06      bne  $f802
f7fc  a5 4e      lda  $4e
f7fe  85 31      sta  $31
f800  a4 4f      ldy  $4f
;     nächstes Byte holen
f802  b1 30      lda  ($30),y
f804  29 c0      and  #$c0
f806  2a         rol
f807  2a         rol
f808  2a         rol
f809  05 57      ora  $57
f80b  85 57      sta  $57
;     codieren und
f80d  b1 30      lda  ($30),y
f80f  29 3e      and  #$3e
f811  4a         lsr
;     zwischenspeichern
f812  85 58      sta  $58
f814  b1 30      lda  ($30),y
f816  29 01      and  #$01
f818  0a         asl
f819  0a         asl
f81a  0a         asl
f81b  0a         asl
f81c  85 59      sta  $59
f81e  c8         iny
f81f  b1 30      lda  ($30),y
f821  29 f0      and  #$f0
f823  4a         lsr
f824  4a         lsr
f825  4a         lsr
f826  4a         lsr
f827  05 59      ora  $59
f829  85 59      sta  $59
f82b  b1 30      lda  ($30),y
f82d  29 0f      and  #$0f
f82f  0a         asl
f830  85 5a      sta  $5a
f832  c8         iny
f833  b1 30      lda  ($30),y
f835  29 80      and  #$80
f837  18         clc
f838  2a         rol
f839  2a         rol
f83a  29 01      and  #$01
f83c  05 5a      ora  $5a
f83e  85 5a      sta  $5a
f840  b1 30      lda  ($30),y
f842  29 7c      and  #$7c
f844  4a         lsr
f845  4a         lsr
f846  85 5b      sta  $5b
f848  b1 30      lda  ($30),y
f84a  29 03      and  #$03
f84c  0a         asl
f84d  0a         asl
f84e  0a         asl
f84f  85 5c      sta  $5c
f851  c8         iny
f852  d0 06      bne  $f85a
f854  a5 4e      lda  $4e
f856  85 31      sta  $31
f858  a4 4f      ldy  $4f
f85a  b1 30      lda  ($30),y
f85c  29 e0      and  #$e0
f85e  2a         rol
f85f  2a         rol
f860  2a         rol
f861  2a         rol
f862  05 5c      ora  $5c
f864  85 5c      sta  $5c
f866  b1 30      lda  ($30),y
f868  29 1f      and  #$1f
f86a  85 5d      sta  $5d
f86c  c8         iny
f86d  84 34      sty  $34
;     Bytes aus
;     Zwischenspeicher
f86f  a6 56      ldx  $56
;     holen, gemäß Tabelle
;     übersetzen und ab $52
;     bis $55
f871  bd a0 f8   lda  $f8a0,x
f874  a6 57      ldx  $57
f876  1d c0 f8   ora  $f8c0,x
;     speichern für Übergabe
;     1. Byte
f879  85 52      sta  $52
f87b  a6 58      ldx  $58
f87d  bd a0 f8   lda  $f8a0,x
f880  a6 59      ldx  $59
f882  1d c0 f8   ora  $f8c0,x
;     2. Byte
f885  85 53      sta  $53
f887  a6 5a      ldx  $5a
f889  bd a0 f8   lda  $f8a0,x
f88c  a6 5b      ldx  $5b
f88e  1d c0 f8   ora  $f8c0,x
;     3. Byte
f891  85 54      sta  $54
f893  a6 5c      ldx  $5c
f895  bd a0 f8   lda  $f8a0,x
f898  a6 5d      ldx  $5d
f89a  1d c0 f8   ora  $f8c0,x
;     4. Byte
f89d  85 55      sta  $55
f89f  60         rts
;     Tabelle dient der
;     Decodierung der GCR-
;     Bytes in die entspr.
;     Hex-Werte.
;     Die $FF-Bytes geben
;     die Positionen der
;     illegalen Bitkombina-
;     tionen an. Sie veran-
;     lassen ggf. einen
;     »24, READ ERROR«.
f8a0  ff ff ff ff ff ff ff ff
f8a8  ff 80 00 10 ff c0 40 50
f8b0  ff ff 20 30 ff f0 60 70
f8b8  ff 90 a0 b0 ff d0 e0 ff
f8c0  ff ff ff ff ff ff ff ff
f8c8  ff 08 00 01 ff 0c 04 05
f8d0  ff ff 02 03 ff 0f 06 07
f8d8  ff 09 0a 0b ff 0d 0e ff
Pufferinhalt decodieren
;     Pufferzeiger auf Null
f8e0  a9 00      lda  #$00
;     setzen
f8e2  85 34      sta  $34
f8e4  85 2e      sta  $2e
;     2. Pufferzeiger
f8e6  85 36      sta  $36
;     Adresse für
;     Ausweichpuffer
f8e8  a9 01      lda  #$01
f8ea  85 4e      sta  $4e
;     nach $4e/4f
f8ec  a9 ba      lda  #$ba
f8ee  85 4f      sta  $4f
;     Pufferadresse Hi retten
f8f0  a5 31      lda  $31
f8f2  85 2f      sta  $2f
;     Bytes decodieren
f8f4  20 e6 f7   jsr  $f7e6
;     1. Byte als Kennzeichen
f8f7  a5 52      lda  $52
;     für Datenblock
f8f9  85 38      sta  $38
f8fb  a4 36      ldy  $36
;     restliche Bytes in
;     Puffer
f8fd  a5 53      lda  $53
f8ff  91 2e      sta  ($2e),y
f901  c8         iny
f902  a5 54      lda  $54
f904  91 2e      sta  ($2e),y
f906  c8         iny
f907  a5 55      lda  $55
f909  91 2e      sta  ($2e),y
f90b  c8         iny
f90c  84 36      sty  $36
;     nächste Kolonne
;     codieren
f90e  20 e6 f7   jsr  $f7e6
f911  a4 36      ldy  $36
f913  a5 52      lda  $52
;     Bytes in Puffer
;     schreiben
f915  91 2e      sta  ($2e),y
f917  c8         iny
f918  f0 11      beq  $f92b
f91a  a5 53      lda  $53
f91c  91 2e      sta  ($2e),y
f91e  c8         iny
f91f  a5 54      lda  $54
f921  91 2e      sta  ($2e),y
f923  c8         iny
f924  a5 55      lda  $55
f926  91 2e      sta  ($2e),y
f928  c8         iny
;     restliche Bytes ggf.
;     codieren
f929  d0 e1      bne  $f90c
;     Prüfsumme Datenblock
;     übernehmen
f92b  a5 53      lda  $53
f92d  85 3a      sta  $3a
;     Pufferadresse Hi
;     zurückholen
f92f  a5 2f      lda  $2f
f931  85 31      sta  $31
f933  60         rts
```

**Listing 3. Die DOS-Routinen zur GCR-Codierung (Schluß)**

## Hier gibt's Clubs

**Luxemburg**

**Commodore Club Luxemburg (C.C.L.)**, 1 Rue des Muguets, L 5970 Itzig/Luxemburg, Clubmagazin, Extrainfos, Musikteil, Kummertelefon;

**Holland**

**Copy Tronics**, Postbus 700, 7400 AS Deventer Nederland, Clubzeitschrift, Clubtreffen;

**Österreich**

**CHIPSI-Computer-Club**, Jasomirgottstr. 3, Mezzanin, 1010 Wien, Programm- und Literaturtausch, Programmierkurse, begünstigter Bezug von Computern, praktische Programmiertips, Clubzeitung, Clubtreffen;

**Wiener Hard- u. Software Computerclub**, René Fürst, Herbortgasse 22, 1110 Wien, Softwareerstellung und Tausch, Hardwareunterstützung, Beitrag 150 Schilling pro Quartal, Beitrittsgebühr 1x150 Schilling, Tel. (0222) 7439154, (0222) 612683

**Bund Technischer Amateure**, Stromstr. 36-38/29/1, 1200 Wien, Clubtreffen, Grundlagen, Tips und Tricks, PRG-Bibliothek, Softwaretausch;

**COMUSCLUS**, Computer User Club Salzburg, A-5033 Salzburg, Postfach 133, Clubtreffen, Clubstammtisch, monatliche Zeitschrift, Buch- und Softwarebibliothek, Arbeitskreise, Programmierkurse, auch für Nichtmitglieder, Sammelbestellungen;

**Ruf-Commodore-User-Club**/Salzburg, A-5020 Salzburg, Haunspergstraße 21, Tel. (0662) 75770, Herr Wiedermann

Ruf-Commodore-User-Club/Graz, Grazbachgasse 60, Tel. (0316) 700266, Herr Hiess

Ruf-Commodore-User-Club/Klagenfurth, A-9020 Klagenfurt, Sponheimer Straße 9-11, Tel. 04222) 55105, Herr Krassnitzer Clubtreffen (wöchentlich), Softwarebibliothek, Hardwareangebot, Programmierkurse etc.

**IKC, Internationaler Kommunikations-club**, Postfach 18, CH-6312 Steinhausen

Info gegen 3 DM, Clubzeitschrift, Beratertelefon, DFÜ, Sektionen, Kurse, Listingkorrekturservice, Meisterschaften, etc. Aktiv-, Passiv- und Ehremitglieder sowie Gönner.

**Schweiz**

**COBEC Computer-Benutzer-Club Wohlen**, Postfach 10, CH-5613 Hilfikon AG, Schweiz, Clubtreffen, Softwarebibliothek, diverse Hardware, Kurse;

**C 64-Anwenderclub Zürich (ACZ)**, Postfach 194, 8029 Zürich, Clubtreffen, Softwarebibliothek, Hardware, Modem, Assembler- und Basic-Kurse, Jahresbeitrag 50 Franken;

**C 64 Club, J. Reich**, Am Glattbogen 170, 8050 Zürich für Anfänger, Clubtreffen, Zeitschrift (geplant), Kurse, Seminare, Softwaretausch, vergünstigter Bezug von Soft- und Hardware, Beitrag 30 Franken

**Syntax Error C 64**, Postfach 253, CH-9004 St. Gallen, Clubtreffen, Mailbox, DFÜ, Softwarebibliothek, Hard- und Softwareentwicklung.

# Memory Map mit Wandervorschlägen

## Diesmal kommen die Adressen 144 bis 152 an die Reihe, die beim C 64 und VC 20 für Kassetten- und Ein-/Ausgabe-Operation verantwortlich sind.

Zu Beginn möchte ich heute einen Nachtrag zu den in Ausgabe 4/85 behandelten Speicherzellen 65/66 bringen. Das kleine Demonstrationsprogramm zur Änderung der Reihenfolge von DATA-Befehlen ist zwar richtig, kann aber unter ganz bestimmten Umständen zu Fehlern führen. Darauf hat mich ein aufmerksamer Leser hingewiesen — ich gebe daher seinen Leserbrief im folgenden direkt wieder. Die Adressen gelten nur für den C 64. Beim VC 20 ist der Sachverhalt aber derselbe.

### Korrektur zu Adressen 65 bis 66

Leserbrief von Martin Kröger aus Kiel:

Bei der Behandlung der Adresse 65/66 hat Dr. H. Hauck auf einen Trick hingewiesen, mit dem der Befehl »RESTORE Zeilennummer« simuliert werden kann. Die Idee ist gut — doch die Ausführung hat einen kleinen, aber entscheidenden Fehler: Die Adresse 61/62 darf nicht mit zwei Befehlen, sondern muß mit einem Befehl ausgelesen werden, da bei einem möglichen Page-Wechsel zwischen den zwei Befehlen der Zeiger nicht verbogen, sondern abgeknickt wird.

Was passiert in der ersten Zeile des Demo-Programms?

10 A1 = PEEK(61):B1 = PEEK(62)

Mit »A1 = PEEK(61)« wird der Variablen A1 der Wert des LOW-Bytes des Zeigers 61/62 zugewiesen. Dieser zeigt am Anfang einer Zeile auf das 0-Byte vor der Linkadresse (hier 2048), so daß A1 den Wert (2048 AND 255) = 0 erhält. Mit »B1 = PEEK (62)« wird der Variablen B1 der Wert des High-Bytes des Zeigers 61/62 zugewiesen. Dieser zeigt aber inzwischen auf das Trennzeichen (»:«) zwischen den beiden Befehlen (hier 2061), so daß B1 den Wert (INT(2061/256)) = 8 erhält. Als Zeiger auf das aktuelle DATA-Element erhalten wir die erwartete Adresse (A1 + B1*256) = 2048.

Was aber, wenn Zeilenanfang und Trennzeichen nicht in derselben Page liegen? Dazu setzen Sie bitte den Basic-Anfang um eine Stelle zurück:

POKE 43,0:POKE 2047,0:NEW

Die Zeiger auf den Zeilenanfang und das Trennzeichen werden dadurch ja ebenfalls verändert, so daß A1 jetzt den Wert (2047 AND 255) = 255 und B1 den Wert (INT(2060/256)) = 8 erhält. Als Zeiger auf das aktuelle DATA-Element erhalten wir nun die völlig unbrauchbare Adresse (A1 + B1*256) = 2303.

Das korrigierte Demo-Programm könnte wie folgt aussehen:

```
10  A1=PEEK(61)+PEEK(62)*256
20  DATA DAS IST DIE 1. ZEILE
30  A2=(61)+PEEK(62)*256
40  DATA DAS IST DIE 2. ZEILE
50  A3=PEEK(61)+PEEK(62)*256
60  DATA DAS IST DIE 3. ZEILE
70  POKE 65,A3 AND 255:
    POKE 66,A3/256:
    READ A$:PRINT A$
80  POKE 65,A1 AND 255:
    POKE 66,A1/256:
    READ A$:PRINT A$
90  POKE 65,A2 AND 255:
    POKE 66,A2/256:
    READ A$:PRINT A$
```

Ich freue mich über Leserbriefe, wie diesen von Herrn Kröger, zeigen sie mir doch, daß meine Erklärungen gelesen und verdaut werden. Und was kann mir besseres passieren, als auf meine eigenen Schlampereien hingewiesen zu werden! Ich habe nämlich den Trick der DATA-Zeilen nicht selbst erfunden, sondern von Sheldon Leemon (eine meiner üblichen Quellen) kritiklos übernommen — und das war falsch.

So, nach diesem Rückblick geht es weiter mit unserer Wanderung. Heute habe ich wieder ein paar Besonderheiten zu bieten.

### Adresse 144 ($90)

**Statusvariable ST**

Diese Adresse enthält ein Byte, welches mit der Statusvariablen ST von Basic identisch ist. Diese reservierte Variable ist im nebenstehenden Texteinschub 1 »ST-atus« näher beschrieben.

Alle Routinen des Betriebssystems, die mit Ein- und Ausgabe zu tun haben, benutzen diese Speicherzelle zum Abspeichern und Abfragen des Status der Ein-/Ausgabeoperationen.

Genauer gesagt, alle Ein-/Ausgabeoperationen, die mit der Datasette und mit dem Floppy-Gerät beziehungsweise dem Drucker zu tun haben, benutzen die Adresse 144. Im Fachjargon sprechen wir vom Kassettenport und vom seriellen Port.

Der dritte Anschluß des Computers nämlich der RS232 oder User-Port benutzt für den Status die Speicherzelle 663. Jedes Bit der Zelle 144 hat eine eigene Bedeutung.

| Kassette: | | |
|---|---|---|
| Bit 2 | (Wert 4) | Kurzer Block |
| Bit 3 | (Wert 8) | Langer Block |
| Bit 4 | (Wert 16) | Lesefehler (nicht korrigierbar) |
| Bit 5 | (Wert 32) | Prüfsummenfehler |
| Bit 6 | (Wert 64) | File-Ende |
| Bit 7 | (Wert 128) | Band-Ende |
| **Floppy/Drucker:** | | |
| Bit 0 | (Wert 1) | Fehler beim Schreiben |
| Bit 1 | (Wert 2) | Fehler beim Lesen |
| Bit 6 | (Wert 64) | Daten-Ende |
| Bit 7 | (Wert 128) | »Device Not Present«-Fehler |

Alle nicht aufgeführten Bits sind nicht benutzt.

Diese Speicherzelle beziehungsweise die Statusvariable ST kann recht nützlich sein. Einige Kochrezepte dafür werden im Texteinschub 1 behandelt.

### Adresse 145 ($91)

**Zwischenspeicher für Abfrage der STOP-Taste**

In den Bildern 1 und 2 ist dargestellt, wie die Tasten des VC 20 und des C 64 miteinander über eine Matrix verbunden sind.

Sechzig mal in der Sekunde unterbricht der Computer seine Arbeit, merkt sich, wo er gerade ist und fragt dann, unter anderem, ob die STOP-Taste gedrückt worden ist. Dadurch wird erreicht, daß die STOP-Taste jederzeit Priorität hat.

Die Abfrage geht so vonstatten, daß das Betriebssystem über das im Bild 1 und 2 gezeigte Spaltenregister 56320 (beim VC 20: 37152) diejenige Tastenspalte anwählt, in welcher sich die STOP-Taste befindet. Aus Bild 1 und 2 sehen wir, daß dies die Spalte mit der Codenummer 127 beziehungsweise 247 ist. Ist in dieser Spalte eine Taste gedrückt, wird an ihrer Stelle eine Null in das Auslese-Register 56321 (VC 20: 37153) geschrieben. Die dadurch entstandene Dualzahl wird in die Speicherzelle 145 gebracht.

Es ist sicher verständlich, daß auf diese Weise nicht nur die STOP-Taste, sondern alle Tasten der Spalte 127 (247) abgefragt werden können. Ein kleines Demonstrationsprogramm kann das beweisen:

```
10 PRINT PEEK (656321); PEEK
(145)
20 GOTO 10
```

Beim VC 20 ist statt 56321 natürlich 37153 einzusetzen.

Das Zahlenband kann durch die Tasten der genannten Spalte — und nur durch diese — beeinflußt werden.

### Adresse 146 ($92)

**Zeitkonstante beim Lesen vom Band**

Die Speicherzelle enthält eine vom Betriebssystem einstellbare Zahl, welche die kleinen Unterschiede in der Aufnahmegeschwindigkeit ausgleicht, die bei verschiedenen Datasetten vorkommen können.

Diese Zeitkonstante steht im Zusamenhang mit der Zahl, die in den Speicherzellen 176/177 steht.

Eine Veränderung der Konstante in Basic ist nicht möglich.

### Adresse 147 ($93)

**Flagge für LOAD oder VERIFY**

Diese Flagge dient dem Betriebssystem, um zu unterscheiden, ob eine LOAD-Operation nur LOADen oder aber VERIFYen soll.

Sie ist identisch mit der Flagge des Basic-Übersetzers in Speicherzelle 10. Genauere Hinweise bitte ich der Beschreibung von Zelle 10 in Ausgabe 12/84, Seite 132 zu entnehmen.

### Adresse 148 ($94)

**Flagge für Floppy/Drucker Ausgang**

Das Betriebssystem benutzt diese Speicherzelle, um anzuzeigen, daß ein Zeichen im Ausgabepuffer steht, welches zum Floppy-Gerät oder zum Drucker geleitet werden soll. Diese Flagge setzt alle am seriellen Port angeschlossenen Geräte in den Zustand »Listen«, das heißt bereit zu sein, Daten aufzunehmen.

### Adresse 149 ($95)

**Zeichen im Ausgabepuffer**

In dieser Speicherzelle wird das Zeichen abgelegt, welches als nächstes über den Serial Port zum Floppy-Gerät oder zum Drucker transportiert wird, sobald die Flagge in 148 die Bereitschaft anzeigt.

### Adresse 150 ($96)

**Arbeitsspeicher für die Band-Leseroutine**

Diese Speicherzelle wird zur Zwischenspeicherung von Da-


64ER ONLINE

ten beim Lesen einer Kassette benutzt.

## Adresse 151 ($97)

**Zwischenspeicher des X-Registers**

Maschinen-Programmierer kennen das X-Register des Mikroprozessors. Beim Lesen eines Zeichens von der Datasette wird der Inhalt des X-Registers in dieser Adresse zwischengespeichert.

## Adresse 152 ($98)

**Anzahl der offenen Files**

Ein File, oder auf Deutsch gesagt, eine Datei, wird mit dem Befehl OPEN eröffnet. Nach OPEN folgt die Nummer der Datei; sie ist beliebig wählbar bis maximal 255. Als zweites folgt die Nummer des Gerätes, mit dem die Verbindung hergestellt werden soll.

Es ist erlaubt, mehrere Dateien gleichzeitig geöffnet zu halten, vorausgesetzt die Nummern der Dateien sind verschieden.

In Speicherzelle 152 wird festgehalten, wieviel Dateien gleichzeitig geöffnet sind. Dieses kleine Programm zeigt es uns deutlich:

```
10   FOR K=10 TO 22
20   PRINT PEEK (152),K
30   OPEN K,0
40   NEXT K
```

Mit der FOR...NEXT-Schleife der Zeilen 10 und 40 eröffnen wir 13 Dateien hintereinander, und zwar — wie Zeile 30 uns deutlich macht — mit der Tastatur. Die Tastatur hat die Nummer 0, der Drucker die Nummer 4, das Floppy-Gerät die Nummer 8 und die Datasette die Nummer 1. Ich habe die Tastatur gewählt, obwohl es keinen Sinn ergibt, weil sie die vielen Eröffnungen ohne zu unterbrechen akzeptiert.

Nach RUN sehen wir links untereinander den Inhalt von 152, also die Anzahl der eröffneten Dateien. Rechts steht jeweils die Nummer der eröffneten Datei.

Nach der 10. Datei bricht das Programm ab und druckt uns die Fehlermeldung TOO MANY FILES aus.

Das heißt es sind gleichzeitig nur 10 Dateien betreibbar. Wenn wir oben in Zeile 10 die Zahl 22 durch 19 ersetzen, läuft das Programm fehlerfrei.

Eine Datei, die unter einer bestimmten Nummer eröffnet ist, kann nicht noch einmal eröffnet werden. Fügen Sie bitte dem Programm noch die folgende Rücksprungzeile hinzu:

```
50   GOTO 10
```

In der 10. Zeile sehen wir jetzt die 10 als Inhalt von 152 und als

## Texteinschub 1

**ST-atus**

Neben den Befehlen (wie PRINT) und den Funktionen (wie COS) hat Basic auch noch drei fest definierte Variable, nämlich TI, TI$ und ST.

Von den dreien ist ST wohl am seltensten anzutreffen, Grund genug, hier ein wenig darüber zu berichten.

Der Anlaß ist natürlich, daß der Wert von ST immer in der Speicherzelle 144 steht, die ja heute in der Memory Map vorkommt.

Bei der Beschreibung wurde schon erwähnt, daß ST den Status nach der letzten Ein- beziehungsweise Ausgabeopration angibt, beschränkt allerdings nur auf Operationen mit der Datasette und der an einem gemeinsamen Ausgang angeschlossenen Floppy und Drucker.

Dementsprechend zeigt die Tabelle bei der Speicherzelle 144 diese beiden Fälle.

Wichtig ist noch zu erwähnen, daß nicht nur die in der Tabelle gezeigten Zahlen für ST auftreten, sondern auch Kombinationen davon. So ergibt zum Beispiel ein zu kurzer Block (4) und ein gleichzeitig aufgetretener Prüfsummenfehler (32) ein Wert von 36.

**Kassettenoperationen**

Zuerst testen wir mit einem Datei-Programm auf »File-Ende«. Geben Sie bitte folgendes Programm ein:

```
10   OPEN 1,1,1,"DATEI"
20   PRINT#1, "QWERTY"
30   CLOSE 1
40   END
```

Zur Erinnerung: nach dem OPEN-Befehl folgt zuerst die Nummer der Datei (ich nehme hier 1), dann die Gerätenummer (1 = Datasette) und schließlich die Sekundäradresse (1 = Schreiben).

Jetzt kommt der Lesevorgang:

```
50   OPEN 2,1,0"DATEI"
60   FOR K=1 TO 10
70   GET#2,A$
80   PRINT A$;ST
90   NEXT K
100  CLOSE 2
```

In Zeile 50 eröffnen wir wieder eine Datei (diesmal Nummer 2) für die Datasette, jetzt aber zum Lesen (Sekundäradresse = 0). Die Schleife der Zeilen 60 und 90 schreiben uns 10 mal ein Zeichen (A$) und den Wert von ST auf den Bildschirm.

Jetzt geht es los. Mit RUN starten wir den ersten Teil des Programms. Nach dem Schreibvorgang und der READY-Meldung (nach Zeile 40) müssen Sie das Band zurückspulen und mit GOTO 50 ab Zeile 50 weiterfahren. Jetzt wird die Datei gelesen.

Wir erhalten untereinander die sechs Buchstaben von Zeile 20, daneben für ST lauter Nullen. Am Ende allerdings erscheint eine 64. Das ist der in der Tabelle angegebene Wert von ST für »File-Ende«.

Da die FOR-NEXT-Schleife zu lang ist, schießen wir mit dem Lesen über das File-Ende hinaus. Normalerweise kennen wir natürlich die Länge einer Datei nicht. Deshalb ist es besser mit GOTO zurückzuspringen und das File-Ende abzufragen.

Löschen Sie bitte Zeile 60 und 90 und fügen Sie als Rücksprung und Prüfung ein:

```
85   IF ST=64 THEN 100
90   GOTO 70
```

Statt nach ST können wir natürlich genau so gut nach PEEK(144) fragen. Ein erneutes GOTO 50 bringt das erwünschte Resultat.

Um den vorhin schon erwähnten »kurzen Block« zu sehen, müssen wir einen entsprechenden Fehler künstlich erzeugen.

Löschen Sie bitte den ersten Teil des Programms bis einschließlich Zeile 40. Wir behalten also nur den Leseteil ab Zeile 50. Dann laden wir dieses Programm (Band vorher am besten wieder zurückspulen) mit SAVE "DATEI" nicht als Datei, sondern als ganz gewöhnliches Programm. Wenn es geladen ist, bitte das Band wieder zurückspulen.

Mit RUN starten wir jetzt das Lese-Programm, welches eine Datei sucht, aber nur ein Programm findet, allerdings mit dem richtigen Namen. Natürlich findet es einen Fehler und wir erhalten als Ausdruck:

```
36   oder manchmal auch 4
64                      64
```

Die Zahl 36 entsteht aus 32 + 4, das bedeutet Prüfsummenfehler + Block zu kurz. Danach folgt wie vorher das File-Ende.

Die normale Blocklänge entspricht der Länge des Bandpuffers, in den die Datasette einspeichert. Er ist 191 Bytes lang. In unserem Fall war offenbar der Block nicht ganz voll.

Der Prüfsummenfehler tritt dann auf, wenn eine der Überprüfungen von Kassettenoperationen einen Fehler gefunden hat. Der Blockfehler, auch der des zu langen Blocks, interessiert wohl weniger. Aber ein durch die Prüfungen gefundener Fehler könnte, frühzeitig noch vor dem Ausstieg des Programms entdeckt, abgefangen und ausgenutzt werden.

Die Formel dafür, ins obige Program eingebaut ist:

85 IF ST<64 OR ST>8 THEN..(zum Beispiel LIST)

Statt LIST kann man natürlich auch etwas anderes nehmen.

**Floppyoperationen**

Bei der Floppy bedeutet ST = 64, »Daten-Ende«, das ist etwa dasselbe wie bei der Datasette. Um es zu überprüfen nehmen wir dasselbe Programm wie vorher, nur müssen wir die Datei-Zeilen der Floppy anpassen. Das sieht dann so aus:

```
10   OPEN 1,8,1,"DATEI,S,W"
20   PRINT#1,"QWERTY"
30   CLOSE 1
40   END

50   OPEN 2,8,0,"DATEI,S,R"
60   FOR K=1 TO 10
70   GET#2,A$
80   PRINT A$;ST
90   NEXT K
100  CLOSE 2
```

Das Ergebnis sieht hier so aus:

```
64
66
66
```

Die 64 ist natürlich wie erwartet der »Wert« für Daten-Ende. Die 66 ist 64 + 2, entstanden dadurch, daß wir über das Datenende hinausgelesen haben. Die 2 bedeutet »Fehler beim Lesen« (in den englischen Beschreibungen heißt es »Read Time Out«). Ähnliches gilt für ST = 1, das bedeutet »Fehler beim Schreiben« (englisch: Write Time Out), nur weiß ich leider nicht, wie es vorzuführen ist. Wie bei der Datasette kann das Überlesen natürlich mit der Abfrage IF ST = 64 THEN...und GOTO... gestoppt werden.

Interessant ist noch der Status beim Fehler »DEVICE NOT PRESENT«, den wir dadurch erzeugen, daß wir ein Programm, oder die Directory, aus der Floppy laden wollen, ohne daß dieses Gerät angeschlossen oder eingeschaltet ist. Nach der Fehlermeldung geben wir direkt ein:

PRINT ST oder PRINT PEEK(144)

und wir erhalten die Zahl 128.

Wie man allerdings in einem Basic-Programm durch Abfrage von ST = 128 die Fehlermeldung »Device Not Present« und den dann folgenden Programmabbruch vermeiden kann, weiß ich leider auch nicht. Die einzige Möglichkeit, die ich kenne, verwendet die sogenannte Interrupt-Methode. Diese ist von Christoph Sauer im letzten Teil seines Kurses »Der gläserne VC 20« in Heft 3/85, Seite 155, beschrieben worden.

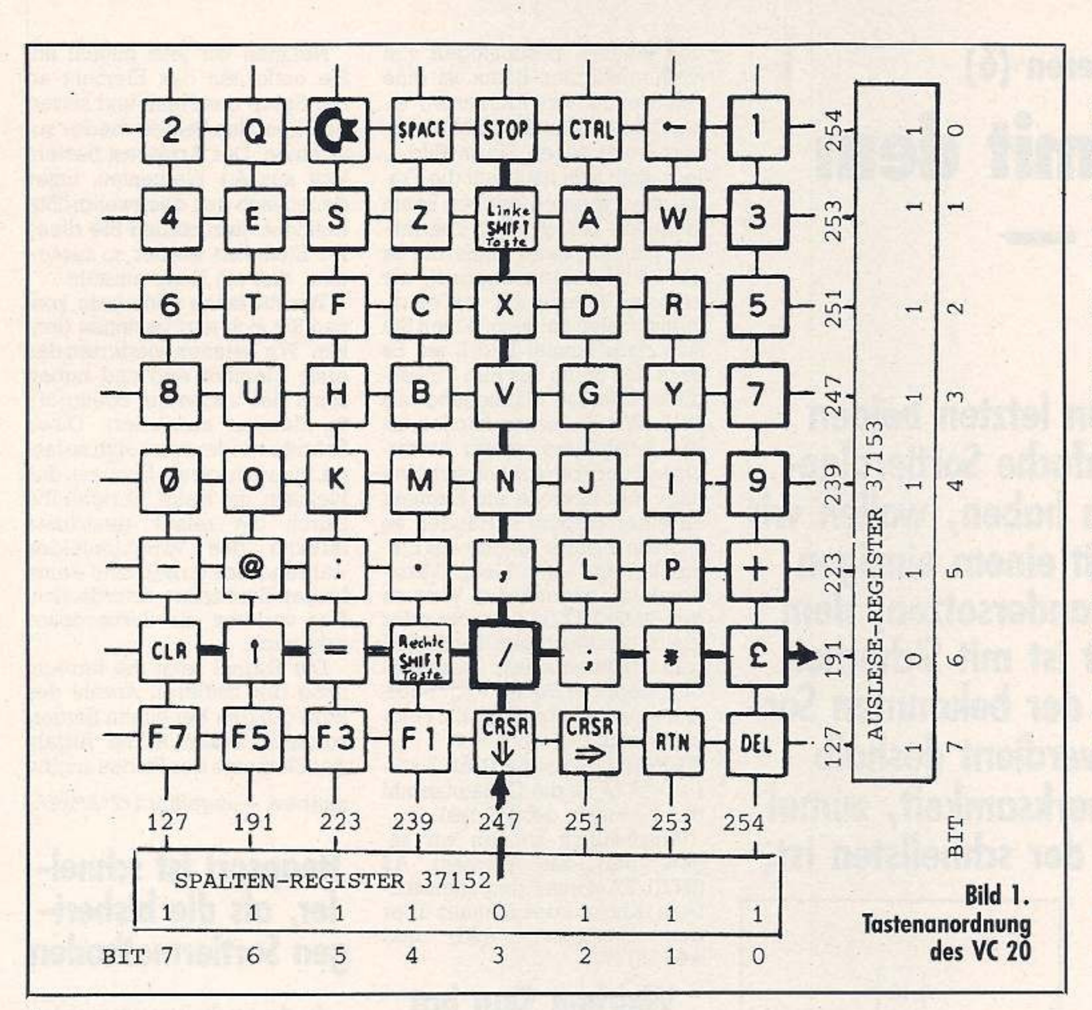



**Bild 1. Tastenanordnung des VC 20**

Sie werden vielleicht fragen, warum ich das so ausführlich beschreibe. Nun, in einem Programm kann es sicher sehr nützlich sein, die Zelle 152 mit PEEK nach der Datei-Lage abzufragen und entsprechend Maßnahmen zu treffen, ehe die Fehlermeldung das Programm abbricht.

Mit POKE 152,0 aber müssen Sie aufpassen. Es ersetzt nämlich nicht (!!) den CLOSE-Befehl. Probieren Sie es aus: Um das kleine Programm oben per Drucker auszudrucken, brauchen wir:

OPEN 1,4: CMD 1: LIST

Wenn Sie jetzt die Zeile 152 auf 0 POKEn und dann LIST eintippen, wird trotzdem wieder auf dem Drucker gelistet und nicht auf dem Bildschirm. Die vorgeschriebene Schließmethode mit PRINT #1:CLOSE 1 geht jetzt aber auch nicht mehr, denn das Betriebssystem ist ja im Glauben, daß keine Datei eröffnet ist — schöner Schlamassel!

Erst eine Neueröffnung bringt alles wieder in die Reihe. Also Vorsicht mit der Anwendung der Speicherzelle 152. Eine Möglichkeit, alle Dateien auf einen Schlag zu schließen, gibt es aber doch:

SYS 65511 besorgt das, sowohl beim C 64 als auch beim VC 20.

(Dr. Helmut Hauck/ah)

neue Dateinummer (K) wieder die 10. Das Programm bleibt aber stehen und meldet FILE OPEN. Es hat recht, denn die Datei 10 ist bereits als erste eröffnet, aber nicht wieder geschlossen worden.

Das Betriebssystem macht das so, daß jede der Dateinummern in eine Tabelle geschrieben wird, die in den Speicherzellen 601 bis 610 stehen. Bei jedem OPEN-Befehl wird dort nachgeschaut, ob die Filenummer existiert. Wenn ja wird die Fehlermeldung FILE OPEN ERROR ausgegeben. Bei jedem CLOSE-Befehl wird die entsprechende Nummer aus der Tabelle gelöscht.

Wir können aber auch eine 0 in die Speicherzelle 152 POKEn, wodurch dem Betriebssystem vorgegaukelt wird, daß keine Datei eröffnet ist. Schieben Sie im Programm einfach die Zeile ein:

45 POKE 152,0

und das Programm läuft ewig weiter.

Die Speicherzelle 152 ist also der Wächter über die Anzahl der eröffneten Dateien. Steht sie auf 0, dann wird eine Neueröffnung am Anfang der Tabelle ab 601 eingetragen. Die Tabelle ihrerseits ist der Wächter über Exklusivität der Dateinummern. Ich zeige Ihnen das noch genauer, wenn wir zu 601 kommen.

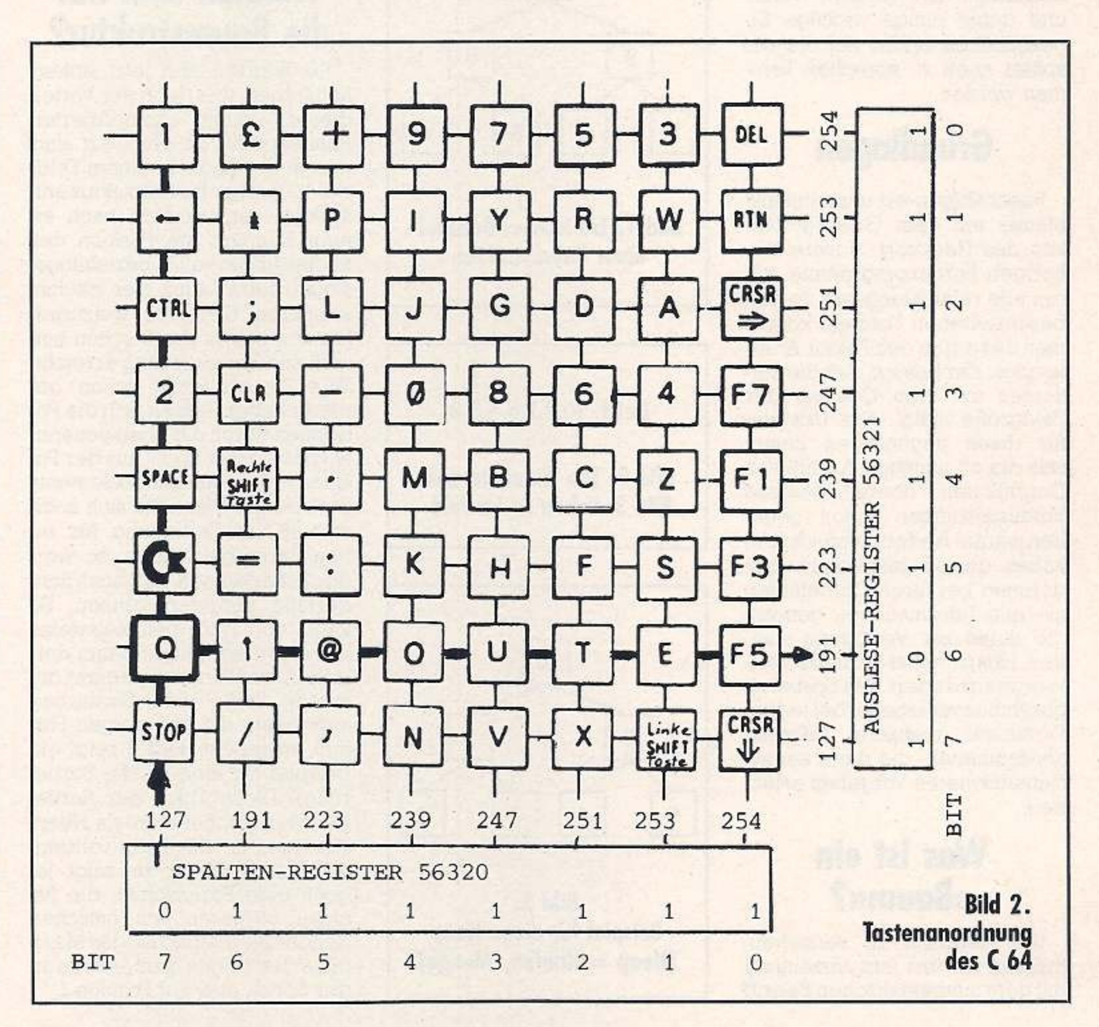



**Bild 2. Tastenanordnung des C 64**

## Effektives Programmieren (6)

# Sortieren mit dem Computer — (Teil 3)

**Nachdem wir in den letzten beiden Folgen mehrere einfache Sortieralgorithmen besprochen haben, wollen wir uns diesmal nur mit einem einzigen Algorithmus auseinandersetzen, dem Heapsort. Heapsort ist mit Sicherheit die komplizierteste der bekannten Sortiermethoden und verdient deshalb unsere ganze Aufmerksamkeit, zumal sie dazu noch eine der schnellsten ist.**

Zuerst einmal zum Ausdruck »heap«. Ein Heap (engl.: Haufen, Menge) bezeichnet ein Variablenfeld, das auf eine ganz bestimmte Art sortiert wurde und damit einige wichtige Eigenschaften erfüllt, auf die wir später noch zu sprechen kommen werden.

## Grundlagen

Beschäftigen wir uns zunächst einmal mit dem Grundgedanken des Heapsort. Unsere bisherigen Sortierprogramme waren alle relativ langsam. In den beschriebenen Formeln konnte man das durch den Faktor $A^2$ erkennen, der besagt, daß die Sortierzeit mit dem Quadrat der Feldgröße steigt. Die Ursache für diese ungünstigen Zeiten war die oft unnötige Anzahl von Durchläufen, obwohl gewisse Voraussetzungen schon gegeben waren. Anders ausgedrückt haben unsere bisherigen Programme bei ihren Durchläufen nie alle Informationen genutzt, die ihnen zur Verfügung standen. Heapsort hat an dieser Stelle etwas geändert. Der Sortieralgorithmus verarbeitet bei jedem Durchlauf mehrere Informationseinheiten, die dann ein systematischeres Vorgehen erlauben.

## Was ist ein »Baum«?

Um Heapsort zu verstehen, müssen wir uns jetzt ersteinmal mit dem mathematischen Begriff des »Baums« beschäftigen. Ein mathematischer Baum ist eine Anordnung von Elementen einer Menge. Ein Beispiel für einen Baum sehen Sie in Bild 1.

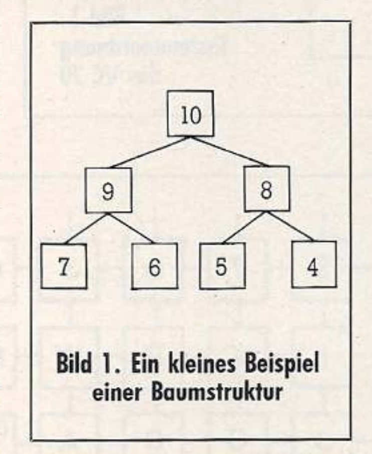



**Bild 1. Ein kleines Beispiel einer Baumstruktur**

Feld: 10,9,7,5,4,3,2

**Bild 2. Die Elemente aus Bild 3 stehen so im Feld**

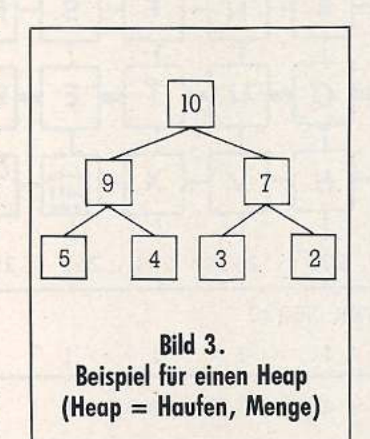



**Bild 3. Beispiel für einen Heap (Heap = Haufen, Menge)**

Jetzt stellt sich natürlich die Frage, wie wir einen solchen Baum in unsere Feldstruktur überführen können, denn immerhin ist ein Baum zweidimensional, wir arbeiten jedoch nur mit eindimensionalen Feldern. Sehen Sie sich dazu einmal Bild 2 an. Es zeigt den Baum aus Bild 3 in eindimensionaler Darstellung als Feld. Wie Sie sehen können, ist die Aufstellung streng hierarchisch und berechnet sich wie folgt: Wir nehmen ein Element an einer Position I. Handelt es sich bei unserer Menge aus Elementen um einen Heap (Variablenfeld), so muß der Wert an der Stelle INT (I/2) größer oder gleich des Wertes an der Stelle I sein. Mathematisch beschrieben sieht die Sache folgendermaßen aus (wobei A$(I) ein Feldelement ist):

A$(INT(I/2)) > = A$(I), für 2 < = I < = A (A ist die Gesamtanzahl der Elemente des Feldes).

Anschaulich können wir sagen, daß das Element A$ (INT(I/2)) immer der nächsthöhere »Knoten« des Baumes über den Elementen A$(I) und A$(I+1) ist.

## Welchen Sinn hat die Baumstruktur?

Sie werden sich jetzt sicherlich fragen, was denn der Vorteil dieser recht komplizierten Baumstruktur ist. Nun, wir sind jetzt in der Lage, in einem Feld, welches einer Baumstruktur entspricht, ganz einfach nach einem Element die Position der nächstgrößeren, beziehungsweise die Position der nächstkleineren Elemente festzustellen und haben damit schon einmal eine Vorsortierung erreicht. Wie Sie vielleicht schon gemerkt haben, lassen sich die Positionen durch die oben genannte Formel ganz leicht aus der Position des Ausgangselements berechnen. Wenn Sie sich noch einmal die Bedingung für einen Baum betrachten, so werden Sie erkennen, daß auch Sonderfälle auftreten können. So kann ein Feld beispielsweise keinen vollständigen Baum darstellen, sondern nur ab einer gewissen Stelle eine Sortierung aufweisen, die der obigen Formel entspricht (Bild 3 zeigt ein Beispiel für eine solche Sortierung). Diese Form der Sortierung bezeichnet man als Heap. Ein Heap ist also nicht vollständig durchsortiert; er zeigt jedoch eine Eigenschaft, die für unser Sortieren von entscheidender Bedeutung ist: das Maximum des Feldes steht jeweils an der Spitze, also auf Position 1.

Nehmen wir jetzt einmal an, Sie entfernen das Element an der Spitze des Heap und setzen den Rest des Feldes wieder zusammen. Der Arrayrest besteht jetzt aus A-1 Elementen, unter denen sich nun das zweitgrößte befindet. Jetzt stellen Sie diese A-1 Elemente wieder so zusammen, daß ein Heap entsteht...

Wie die Reihe weitergeht, können Sie sich jetzt bestimmt denken. Wir nehmen wiederum das erste Element weg und haben somit das insgesamt zweitgrößte Element aussortiert. Diese Schritte wiederholen sich so lange, bis nur noch ein Element, das kleinste, im Heap übrigbleibt. Durch die relativ geordnete Struktur des Variablenfeldes während der Arbeit sind keine langen Suchzeiten erforderlich. Das verkürzt die Sortierdauer erheblich.

Die Formel zeigt die Berechnung der mittleren Anzahl der Bewegungen bei einem Sortiervorgang, wobei A die Anzahl der Elemente des Feldes angibt:

mittbew = ungefähr 1/2*A*ld(A)

## Heapsort ist schneller, als die bisherigen Sortiermethoden

An dieser Stelle zeigt sich bereits der deutliche Unterschied von Heapsort zu den vorher besprochenen Sortiermethoden. Während diese immer den Faktor $A^2$ in ihren Formeln hatten, steigt die Zahl der Bewegungen bei Heapsort im Mittel nur um den Faktor ld (A), wobei ld den Logarithmus zur Basis 2 darstellt. Der Logarithmus ld kann durch den natürlichen Logarithmus (ln) auf die folgende Art dargestellt werden:

ld(A) = ln(A)/ln(2).

An dieser Stelle gleich einmal eine Bemerkung zum Logarithmus: Auf dem C 64 und auf dem VC 20 existiert jeweils nur der natürliche Logarithmus (ln). Im Basic-Dialekt dieser Computer wird jedoch eine unübliche Abkürzung für diesen Logarithmus verwendet. Eine Berechnung erfolgt hier mit dem Befehl LOG, der eigentlich für den Logarithmus der Basis 10 (Zehnerlogarithmus) steht.

Jetzt aber wieder zurück zu Heapsort. In Bild 4 können Sie erkennen, was vorher besprochen wurde. Unser Beispiel-Heap aus Bild 3 wird sortiert, wobei jeweils das erste Element aussortiert und dann der Rest des Arrays wieder zusammengesetzt wird.

Das Programm zu Heapsort ist in Listing 1 dargestellt und benötigt zu seiner Ausführung die beiden anderen Programmteile,


64ER ONLINE

die als Listing 1 und 2 in der 64'er, Ausgabe 4/1985 auf Seite 149 abgedruckt wurden. Um die Abfragepunkte bei Heapsort deutlich darzustellen, wurde in Listing 1 auf FOR-NEXT-Schleifen verzichtet, was allerdings zu Lasten der Geschwindigkeit geht. Wenn wir in der nächsten Ausgabe Quicksort besprechen, werden wir unter anderem auch auf Geschwindigkeitsfaktoren (Garbage-Collection, Vergleiche, Indizierung) eingehen. Außerdem werden wir einen Zeitvergleich über alle bisher besprochenen Routinen durchführen, zu dessen Zweck die Programme natürlich optimiert werden. Bild 5 zeigt einen Ausdruck von Heapsort während der Arbeit. In Bild 6 sehen Sie, wie schon in den vorherigen Folgen, einen Programmablaufplan, anhand dessen es keine großen Probleme bereiten sollte, Heapsort für eigene Anwendungen oder auf andere Programmiersprachen umzuschreiben.

Hiermit wollen wir es für heute aber auch schon bewenden lassen. Heapsort dürfte Sie sicherlich noch einige Zeit beschäftigen, da diese Methode nicht ganz leicht zu durchschauen ist. In der nächsten Folge werden wir uns ausführlich mit dem Quicksort beschäftigen, dem wohl schnellsten Sortieralgorithmus.

(Karsten Schramm/gk)

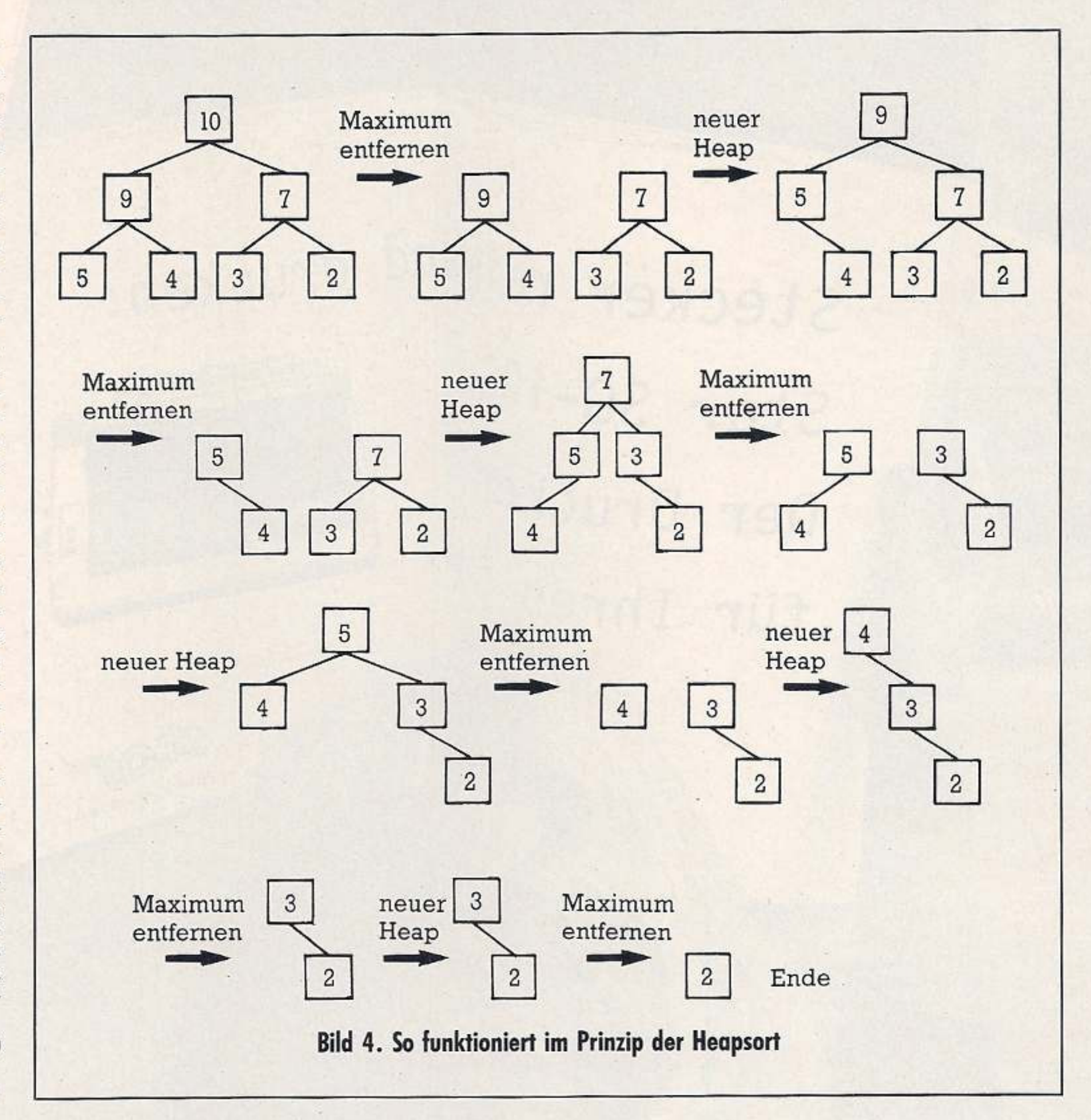



**Bild 4. So funktioniert im Prinzip der Heapsort**

```
WJH AMJ KPB MRB ETQ KOG MOO TFJ JEE UCV
WJH AMJ KPB MRB UCV KOG MOO TFJ JEE ETQ
WJH AMJ KPB TFJ UCV KOG MOO MRB JEE ETQ
WJH AMJ MOO TFJ UCV KOG KPB MRB JEE ETQ
WJH UCV MOO TFJ ETQ KOG KPB MRB JEE AMJ
WJH UCV MOO TFJ ETQ KOG KPB MRB JEE AMJ
UCV TFJ MOO MRB ETQ KOG KPB AMJ JEE WJH
TFJ MRB MOO JEE ETQ KOG KPB AMJ UCV WJH
MRB JEE MOO AMJ ETQ KOG KPB TFJ UCV WJH
MOO JEE KPB AMJ ETQ KOG MRB TFJ UCV WJH
KPB JEE KOG AMJ ETQ MOO MRB TFJ UCV WJH
KOG JEE ETQ AMJ KPB MOO MRB TFJ UCV WJH
JEE AMJ ETQ KOG KPB MOO MRB TFJ UCV WJH
ETQ AMJ JEE KOG KPB MOO MRB TFJ UCV WJH
AMJ ETQ JEE KOG KPB MOO MRB TFJ UCV WJH
---------------------------------------
AMJ ETQ JEE KOG KPB MOO MRB TFJ UCV WJH
 10  ELEMENTE
```

**Bild 5. Ein Feld wird im Heapsort sortiert. Man erkennt, daß sich die Geschwindigkeit des Heapsort erst bei größeren Feldern bemerkbar macht.**

```
10000 REM SORTIEREN MIT BAEUMEN                    <104>
10010 REM                                          <208>
10020 REM HEAPSORT                                 <064>
10030 REM                                          <228>
10040 LG=INT(A/2)+1:RG=A                           <107>
10050 IF RG<=1 THEN 10190                          <197>
10060 IF LG<=1 THEN 10100                          <192>
10070 REM AUFBAU DES HAUFENS                       <166>
10080 LG=LG-1                                      <059>
10090 I=LG:GOTO 10140                              <216>
10100 REM WEGNEHMEN DES MAXIMUMS                   <021>
10110 S$=A$(1):A$(1)=A$(RG):A$(RG)=S$              <137>
10120 RG=RG-1                                      <111>
10130 I=1                                          <229>
10140 X$=A$(I)                                     <240>
10150 P=0: REM FLAG FUER NICHT GEFUNDEN            <214>
10160 IF 2*I<=RG AND P=0 THEN 10200                <002>
10170 A$(I)=X$                                     <014>
10180 GOSUB 3000: GOTO 10050                       <244>
10190 GOTO 10300: REM ENDE                         <087>
10200 REM FELDVARIABLE A$(I) EINORDNEN             <144>
10210 J=2*I                                        <044>
10220 IF J<RG THEN IF A$(J)<A$(J+1)THEN J=
      J+1                                          <188>
10230 IF X$>=A$(J)THEN 10260                       <039>
10240 A$(I)=A$(J)                                  <217>
10250 I=J:GOTO 10160                               <050>
10260 P=1: REM FLAG FUER PLATZ GEFUNDEN            <091>
10270 GOTO 10160                                   <199>
10280 :                                            <138>
10290 :                                            <148>
10300 REM ENDE                                     <015>
```

**Listing 1. Das Programm zum Heapsort (Sortieren mit Baumstruktur) ►**


64ER ONLINE

Bild 6. Dieses Flußdiagramm zeigt noch einmal die Funktionsweise des Heapsort. Damit wird auch die Umsetzung des Algorithmus in andere Programmiersprachen erleichtert.

Fortsetzung von Seite 115

## Einführung in CAD mit dem Commodore C 64

Das rechnerunterstützte Konstruieren (Computer aided Design, kurz CAD) macht Gebrauch von den Fähigkeiten des Computers, zu rechnen, Daten zu verwalten und (auf dem Bildschirm oder dem Grafikdrukker) zu zeichnen, womit hier technisches Zeichnen, nicht Malen gemeint ist. Der Autor dieses neuen Data-Becker-Buches betont zu Beginn, daß im Vergleich zu professionellen Systemen die geringe Bildschirm-Auflösungsfähigkeit des C 64 der einzige echte Nachteil sei; die geringe Speicherkapazität und Rechengeschwindigkeit sei durch geschickte Organisation und Geduld wettzumachen. Und so stellt er Schritt für Schritt eine Reihe von kleinen Bausteinen zusammen (Zeichnen von Punkt, Linie, Rechteck, Kreis, Bogen; Strichlinien, Maßpfeile, Schraffuren), aus denen wiederum Makroelemente (Quader, Pyramide, Prisma, Zylinder, Kegel, Kegelstumpf, Kugel) gebildet werden können.

Techniken wie Vergrößerung, Verkleinerung, Zoomen, Duplizieren, Drehen, Spiegeln und Setzen oder Löschen von Einzelpunkten mit einem »Hires-Cursor« werden ebenfalls behandelt und gehören sicherlich zu den interessantesten und nützlichsten Ausführungen des Buches. So hat man schließlich einen Satz von Bausteinen mit den entsprechenden Programmen in Simons Basic zur Verfügung, mit denen man eigene Aufgaben angehen kann.

In diesem Zusammenhang ist allerdings anzumerken, daß die Lösung der betrachteten Probleme hauptsächlich in den Programmen selbst stattfindet und im Text relativ wenig erklärt wird. Dazu gesellt sich die Bemerkung: »Wer wissen will, wie ein Programm strukturiert ist, kommt auch ohne Erklärungen dahinter, und wer es ohne Erklärungen nicht schafft, wird auch mit ihnen nicht glücklich.« Anstelle von 35 Seiten Programmlisting für eine spezielle Beispielzeichnung wäre es vielleicht angebrachter, einige grundsätzliche Dinge etwas ausführlicher zu erklären; seine Programme für eigene Anwendungen muß der Leser schließlich doch selber schreiben.

Auf dem Wege dazu bietet dieses Buch letztlich aber doch eine Reihe von nützlichen Hilfsmitteln, Anregungen und Lösungsansätzen, so daß man das Buch jedem an technischer Grafik Interessierten durchaus empfehlen kann. (Dr. Wolfgang Bunse)

Info: Werner Heift, Einführung in CAD mit dem Commodore 64, Data Becker, Düsseldorf 1985, 302 Seiten, ISBN 3-89011-067-3, 49 Mark.

# Floppy oder Datasette?

## Die Zeiten, in denen das Floppy-Laufwerk 1541 noch weit über 1 000 Mark gekostet hat, sind längst vorbei. So nach und nach scheint die Datasette verdrängt zu werden.

Ein Computer ohne Massenspeicher ist wie ein Auto ohne Räder. Man kann ihn zwar in Betrieb nehmen, doch kommt man nicht allzu weit damit. Erst Massenspeicher machen den Computer so ungeheuer flexibel. Durch sie hat man die Möglichkeit, Programme und Daten dauerhaft zu konservieren, aber auch die Weitergabe und der Kauf von Software wird erst mit einem solchen Medium möglich.

Zu einem Computer gehört also in jedem Fall ein externer Speicher. Für Homecomputer und speziell für den Commodore ist das die Datasette oder das Floppy-Laufwerk 1541. Doch für welchen Massenspeicher soll man sich entscheiden? Ein gewichtiges Argument fällt sofort, auch für den Laien, ins Auge: der Preis. Die Datasette ist mittlerweile für rund 99 Mark zu haben, während das Floppy-Laufwerk immerhin noch mehr als das sechsfache kostet. Doch der Siegeszug der Floppy-Laufwerke ist nicht aufzuhalten. Für Datasette-Besitzer ein Grund zum Nachdenken.

## Floppy oder Datasette?

Was zeichnet denn nun die Floppy gegenüber der Datasette aus? Das gewichtigste Argument dürfte der Komfort sein. Mit der Floppy zu arbeiten ist wesentlich übersichtlicher und auch schneller als mit dem Kassettenrecorder. Das beginnt mit dem Anlegen eines Inhaltsverzeichnisses, dem Directory. Die Titel aller Programme und Dateien, die auf einer Diskette gespeichert sind, werden in dem Directory mit einem Befehl erkennbar. Es informiert Sie über den Namen, die Programmlänge und ob es sich um ein Programm (PRG) oder um Daten (Dateien) handelt. Dateien werden mit SEQ oder USR (sequentielle) oder REL (relative) gekennzeichnet. Gelöschte Programme oder Dateien erhalten die Kennzeichnung DEL, werden aber nicht mehr angezeigt.

Auch der restliche auf der Diskette verfügbare Platz wird angegeben. Soll ein Programm geladen werden, kann das aus diesem Inhaltsverzeichnis heraus aber auch direkt geschehen. Wird ein Programm gelöscht, so wird der freigewordene Platz ohne Verlust und Lücken wieder zur Verfügung gestellt. Bei der Datasette bestehen diese Möglichkeiten nicht.

Sollte ein Datasette-Besitzer in den Genuß einer nicht von seinem Gerät bespielten Kassette kommen, so kann es vorkommen, daß dessen Programme nicht geladen werden können. Der Tonkopf der anderen Datasette braucht nur etwas anders eingestellt sein und schon wird es problematisch. Manche Kassetten haben ab und zu nicht einwandfrei beschichtete Stellen auf dem Band. Sollte beim Speichern eines Programms solch eine Stelle beschrieben werden, wird es später unmöglich, das Programm wieder zu laden. Datasetten-Besitzer gewöhnen sich deshalb sehr schnell an eine Kontrollfunktion, VERIFY genannt. Hier wird das soeben auf Band gespeicherte Programm noch einmal mit dem Inhalt des Computers verglichen. Das bedeutet schon mehr als doppelten Zeitaufwand (vor dem VERIFY muß ja die Kassette zurückgespult werden). Diskettenbesitzer vergessen diesen Befehl bald. Disketten sind viel sicherer als Kassetten.

Sollten Sie sich zu einem leidenschaftlichen Programmierer entwickeln, und das ist beim C 64 nicht ganz ausgeschlossen, werden Sie nicht an die Programmierung einer Dateiverwaltung herumkommen (wollen). Und jetzt bekommt die Datasette gleich zweimal eins aufs Dach. Turbotape-Besitzer (Turbotape beschleunigt Laden und Speichern von Programmen auf Kassette) können dieses Programm jetzt vergessen, denn Daten von sequentiellen Dateien werden nur mit der normalen Geschwindigkeit geladen. Das gibt Ihnen Zeit für Kaffee- und Zigarettenpausen. Den zweiten Schlag erhalten Sie, wenn Sie mit Ihrem Programmierwissen noch weiter vorangeschritten sind und in Zeitschriften oder Büchern über die Begriffe »Relative Datei« und »Direktzugriffsdatei« stolpern. Überlesen Sie einfach diese Kapitel, denn solche interessanten Dateiformen sind mit der Datasette nicht möglich. Aber selbst wenn Sie keine große Ambitionen als Programmierer haben: Dateiverwaltungsprogramme, die diese Formen der Verwaltung besitzen (und das sind die wirklich guten), erhalten Sie nur auf und für Diskette.

Damit wären wir bei einem weiteren Thema: Softwareangebot. Mit stetig sinkendem Preis des Diskettenlaufwerks und damit dessen wachsender Beliebtheit wird immer mehr Software auf den Markt kommen, die es nur auf Diskette gibt, beziehungsweise nur zusammen mit einem Diskettenlaufwerk funktioniert. Selbst die Literatur, dazu zählen auch Computerzeitschriften, befaßt sich immer mehr mit der Floppy. Auch bei uns in der Redaktion bekommen wir die wachsende Präsenz der 1541-Besitzer zu spüren. Das Verhältnis der Programmeinsendungen mit Diskette zu denen mit Kassette liegt zur Zeit bei ungefähr 7:3, mit sinkender Tendenz für Kassette.

## Floppy im Detail

Obwohl der Preisunterschied zwischen Datasette und Floppy-Disklaufwerk nicht gerade gering ist, darf man nicht vergessen, daß die 1541 zu den preisgünstigsten Diskettenlaufwerken überhaupt zählt. Viele Hersteller von Homecomputern bieten erst gar keins an. Und erst recht findet man keine »intelligenten« Laufwerke. Intelligent deshalb, weil die 1541 einen eigenen Mikroprozessor, den 6502, und ein Disketten-Operations-System (DOS) besitzt.

Die Speicherkapazität der 1541 ist zwar nicht berauschend, aber mit ungefähr 170 KByte pro formatierter 5 ¼-Zoll-Diskette für die meisten Anwendungen voll ausreichend. Dabei können bis zu 144 Einträge ins Directory gemacht werden, eine Zahl, die Sie wohl selten erreichen werden. Auch die Übertragungsgeschwindigkeit von Daten vom Computer zur Floppy ist mit runden 3200 Baud nicht gerade als Hexerei zu bezeichnen. Schuld daran ist nicht so sehr die serielle Datenübertragung, sondern vielmehr das DOS selber. Aber immerhin geht das immer noch rund 10 mal schneller als mit der Datasette.

Fortsetzung auf Seite 131

# Reset-Taster für alle Fälle

**Ein Reset-Taster der herkömmlichen Bauart führt nicht immer zum gewünschten Ergebnis. Abhilfe schafft diese 3-Mark-Lösung, die fast jedes Programm abbricht. Sie müssen den C 64 also nicht immer aus- und wieder einschalten, um ein Programm zu beenden.**

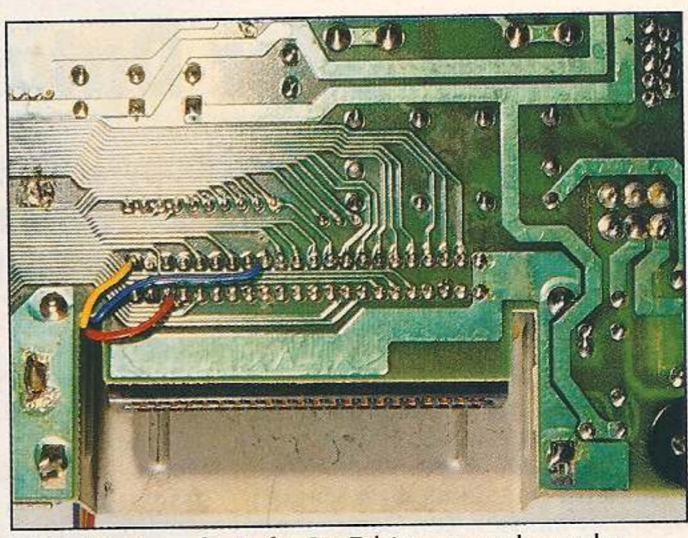


**Bild 1. Die Lösung für Profis. Die Zuleitungen werden an der Platine angelötet. Masse gelb, EXROM blau, Reset rot. Der Expansion-Port wird von rechts nach links numeriert (Aufsicht)**

Ein normaler Reset-Taster für den C 64 kann viele Programme nicht stoppen. Das ist immer dann der Fall, wenn im Programm ein sogenannter Modulstart enthalten ist.

## So funktioniert der Modulstart

Was ist ein überhaupt ein Modulstart? Um das zu erklären, muß man sich das Betriebssystem des C 64 näher betrachten. Nach einem Reset, der auch softwareseitig mit SYS64738 ausgelöst werden kann, sieht sich der Computer den Inhalt der Speicherstellen $8004 bis $8008 an. Steht dort »cbm80« in Form der Commodore-ASCII-Codes, springt der C 64 an die Adresse, die in $8000 und $8001 angegeben ist. Dabei steht in $8000 das Low- und in $8001 das High-Byte des Reset-Vektors. In $8002 und $8003 steht der Warmstart-Vektor, also die Adresse, die durch Drücken von RESTORE angesprungen wird. Steht nun in $8000 und $8001 der Anfang eines Programmes, startet das Programm nach einem Reset immer wieder von neuem. Der Name »Modulstart« kommt daher, daß in jedem Modul, das nach dem Einschalten auch starten soll »cbm80« steht.

Wie Sie sehen, entscheidet also der Inhalt der Speicherzellen $8000-8008 über die Wirksamkeit eines herkömmlichen Reset-Taster, der die Reset-Leitung des C 64 auf Masse legt.

## Wir brechen aus

Da ein Modul die ROM-Konfiguration des C 64 ändert, ist klar, daß kein Reset-Taster ein Modul abschalten kann. Das Modul schließt nämlich eine oder mehrere Leitungen am Expansion-Port kurz und sagt dem C 64 dadurch, welche Speicheraufteilung zu aktivieren ist. Wird die EXROM-Leitung (externes ROM) auf Masse gelegt, schaltet der C 64 den RAM-Speicher von $8000 bis $A000 ab und greift in diesem Bereich auf das ROM des Moduls (EPROM) zu. Genau das hilft uns, um dem Software-Modulstart eines Programmes zu entkommen. Wird nämlich während eines Reset die EXROM-Leitung auf Masse gelegt, sucht der C 64 in einem (nicht vorhandenen) Modul nach der bekannten »cbm80«-Kennung. Da in einem nicht vorhandenen Modul kein »cbm80« steht, durchläuft der C 64 die normale Reset-Routine mit Basic-Kaltstart-Einsprung (Einschaltmeldung). Nach dem Reset wird dann die Verbindung EXROM — Masse aufgehoben und damit die normale Speicherkonfiguration wieder hergestellt. Der C 64 meldet sich mit dem Einschaltbild. Wenn Sie nun einen normalen Reset — etwa mit SYS64738 — auslösen, wird das Programm wieder gestartet. Ob es allerdings noch funktionsfähig ist, hängt vom Programm selbst ab.

## Reset-Schalter in der Praxis

Wie bereits erwähnt, muß während des Reset die EXROM-Leitung auf Masse (GND) gelegt werden. Das macht man am einfachsten mit einem Taster, der zwei Leitungen gleichzeitig schaltet (2 x Ein). Da EXROM während des ganzen Reset Low-Potential haben muß, wird mit einem Kondensator die »Reset-Dauer« begrenzt. Dazu wird ein Elektrolyt-Kondensator mit 16V/47µF in die Leitung vom Reset-Kontakt (Pin C, Expansion-Port) zum Taster geschleift. Minuspol in Richtung Masse. Die Funktion ist einfach zu erklären: Nach Drücken des Tasters liegt auf beiden Kondensatorplatten Masse an, der Reset wird gestartet. Der Kondensator lädt sich nun innerhalb kürzester Zeit auf und die Reset-Leitung liegt wieder auf High-Pegel, während EXROM (Pin 9) noch solange Low-Potential hat bis der Taster losgelassen wird. Damit sich der Elko nach dem Reset wieder entladen kann, überbrückt man ihn mit einem 10kOhm-Widerstand, 1/8 Watt.

Die Resetleitung liegt durch diese »Einschaltbegrenzung« so kurzzeitig auf Masse, daß das Commodore Diskettenlaufwerk 1541 davon vielleicht gar nichts merkt und keinen Reset macht.

## Ein- oder Anbau — das ist hier die Frage

Für den Versierten in Sachen Löten empfiehlt sich der feste Einbau in den C 64 (Garantie beachten). Die Kontakte des Expansion-Ports werden bei dieser Lösung am besten auf der Lötseite der C 64-Platine abgegriffen. In Bild 1 ist Masse an Pin 1 angeschlossen (gelbes Kabel). EXROM an Pin 9 (blaues Kabel) und Reset an Pin C (rotes Kabel). Den Schaltplan finden Sie in Bild 2, die Stückliste in Tabelle 1.

Eine andere Lösung wäre der Anschluß über einen Platinenstecker. Der Stecker muß allerdings selbst gebaut werden. Uns ist keine Be-

zugsquelle bekannt. Zur Steckerfertigung brauchen Sie eine beidseitig kupferkaschierte Epoxidharzplatte mit einer Größe 58 x 50 mm. 58 mm ist die Breite des Expansion-Ports. Bei ausgeschaltetem (!) Computer stecken Sie nun die Platine einige Male in den Expansion-Port und wieder aus, bis die Schleifspuren der Kontakte deutlich zu sehen sind. Zwischen den Spuren wird das Kupfer auf der ganzen Länge der beiden Platinenseiten etwa 1 mm breit abgekratzt. Das kann mit einem Stichel, Körner oder einem spitzen Nagel und einem Lineal als Anschlag geschehen. An die Leiterbahnen können Sie nun die Zuleitungen zum Reset-Taster anschließen. Wer es professionell machen will, kann auch eine kleine Platine speziell dafür ätzen. (hm)

**Stückliste**

Taster 2xEin
Elko 16V/47$\mu$F
Widerstand 10kOhm, 1/8 Watt
Schaltdraht (versch. Farben)
event. beidseitig kupferkaschierte Epoxidharzplatte 58 x 50 mm

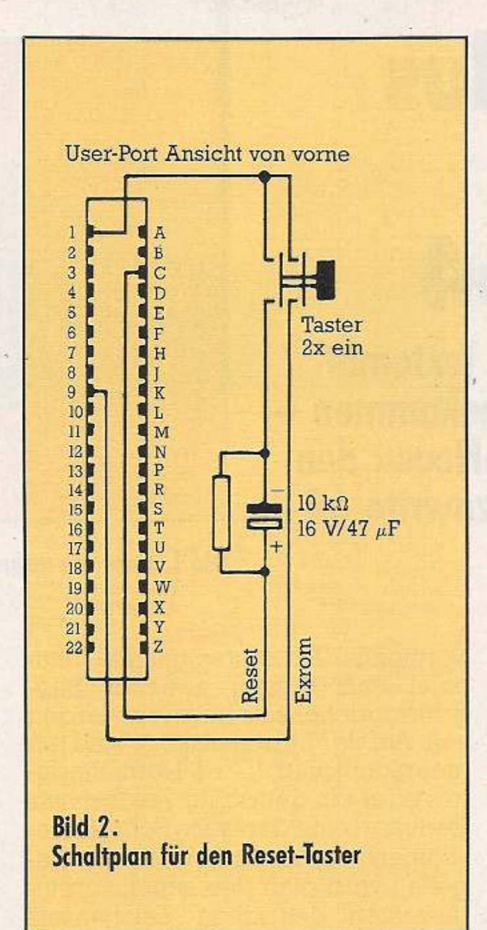



**Bild 2.**
**Schaltplan für den Reset-Taster**

Fortsetzung von Seite 129

## Eine oder zwei Floppies?

Wirklich keine Frage für Leute, die viel mit Dateiverwaltungen oder auch dem Kopieren von Programmen zu tun haben. Denn gerade diese Arbeiten erfordern oft einen Wechsel der Disketten. Mit zwei Floppy-Laufwerken ist dieses zeitraubende und auf die Dauer nervtötende Problem beseitigt. Genaugenommen lassen sich sogar bis zu fünf verschiedene 1541 hintereinander anschließen. Und das Schöne daran ist, daß, ausgenommen natürlich für das Laufwerk selber, keine zusätzlichen Kosten für Interfaces oder sonstige Hardwarezusätze hinzukommen. Einfach mit dem mitgelieferten Kabel an den Computer stecken und fertig. Vielleicht haben Sie jetzt auch den Eindruck, daß die Floppy gegenüber der Datasette doch entscheidende Vorteile hat, auch wenn die Existenzberechtigung der Datasette nicht ganz verschwunden ist. Denn für viele ist letzten Endes die Geldbörse das überzeugendste Argument — jedenfalls bis auf weiteres. (gk)



# Textomat Plus kontra Vizawrite 64

## Der seit langem bekannte Textomat hat einen großen Bruder bekommen — Textomat Plus. Wird der »Neue« den bisherigen Spitzenreiter Vizawrite 64 verdrängen können?

TEXTOMAT PLUS (C) 1985 DATA BECKER GMBH
**** ZEICHENSATZ/DRUCKERSTEUERUNG ****
Druckeransteuerung
Druckerzeichensatz editieren
Ein-/Ausgabe Druckerzeichensatz
Bildschirmzeichensatz editieren
Ein-/Ausgabe Bildschirmzeichensatz
Floppy-Systemoperationen
Tastaturtabelle erstellen
Modem-Parameter
Bedienung des Programms
zurück zu TEXTOMAT PLUS

**Bild 1. Eines von mehreren Menüs beim Textomat Plus.**

Für diejenigen, die unsere Zeitschrift seit einiger Zeit lesen, sind sowohl Textomat als auch Vizawrite 64 bereits alte Bekannte. In Ausgabe 9/84 haben wir den alten Textomat und in Ausgabe 10/84 Vizawrite 64 getestet. Seit damals galt Vizawrite 64 wegen seines ausgereiften und fehlerfreien Konzeptes als bestes Textverarbeitungsprogramm in der Klasse über zweihundert Mark. Für die knappere Kasse konnte man damals den Textomat für 99 Mark ruhigen Herzens empfehlen. Mittlerweile war man in Düsseldorf bei Data Becker nicht müßig und bemühte sich, ein Textverarbeitungsprogramm für den gehobenen Anspruch, in der gleichen Preisgruppe wie Vizawrite 64, zu entwickeln. Seit kurzem ist nun der von W. Schellenberger programmierte Textomat Plus für 248 Mark erhältlich.

Die Einleitung zum Textomat Plus ist vielversprechend — viele Ideen und Anregungen aus dem Kundenkreis sollen beim Textomat Plus berücksichtigt worden sein. Alles in allem soll Textomat Plus in eine neue Dimension der Textverarbeitung führen. Ein hoher Anspruch, den es natürlich zu prüfen galt. Wir haben im Test besonders auf folgende Punkte geachtet:

— Bedienungsfreundlichkeit (sind alle Funktionen einfach erreichbar und sinnvoll plaziert?)
— Arbeitsgeschwindigkeit (wie lange dauert das Erstellen eines Testschriftstückes?)
— Übersichtlichkeit (der Texteingabe, der Textkorrektur, der Sonderfunktionen)

Textomat Plus wird nur auf Diskette geliefert. In alt bekannter Art beginnt kurz nach dem Laden der Weg durch eine Reihe verschiedener Vormenüs. Als erstes möchte Textomat Plus wissen, welcher Bildschirmzeichensatz geladen werden soll. Auf der Masterdiskette sind ein amerikanischer (C 64 Normaltastatur) oder ein deutscher Zeichensatz (simuliert eine deutsche Schreibmaschinentastatur) vorhanden. Im Regelfall wird man den schon voreingestellten deutschen Zeichensatz wählen (Taste F1 drücken), denn Textomat Plus erzeugt auf fast allen Druckern deutsche Sonderzeichen. Im nächsten Menü wird man dazu aufgefordert, den Druckertyp anzugeben. Hat man einen MPS 801, genügt ein Druck auf die Funktionstaste F1. Besitzer eines anderen Druckers müssen hier ihren Druckertyp eintragen und den Rest der Zeile löschen. Im dritten und glücklicherweise letzten Menü kann das Datum in beliebiger Form eingegeben werden. Kurz nach dem Bestätigen mit F1 befindet man sich im Schreibmodus.

## Unterschiedliche Menütechnik

Es ist schon ärgerlich, was man beim Textomat Plus alles anstellen muß, um endlich einen Text eingeben zu dürfen. Flexibilität in allen Ehren, aber wer wechselt seinen Drucker schon täglich? Alle dauerhaften Konfigurationsparameter wie Bildschirmzeichensatz, Druckertyp, Anzahl der Laufwerke und die Bildschirmfarben gehören einmal festgelegt und abgespeichert. Hierzu ist die auch sonst sinnvolle Funktion des Formulars bestens geeignet. Beim Laden des Programms sollte ein vom Anwender erstelltes und unter bestimmten Namen (beispielsweise »Standardformular«) abgespeichertes Formular automatisch mitgeladen werden. Wer dann noch etwas ändern möchte, kann das im Textmodus immer noch tun. Auch die Eingabe des Datums sollte dem Wunsch des Anwenders vorbehalten bleiben. Das wurde auch erkannt, denn im Formular kann jederzeit das Datum eingegeben werden, fragt sich, warum dann noch ein extra Menü kurz nach dem Laden dafür vorgesehen ist?

Das Konzept, das in dieser Hinsicht von Vizawrite 64 verwirklicht wird, gefällt da schon wesentlich besser. Nach dem Start erscheint ein Hauptmenü, von dem aus alle wesentlichen Grundfunktionen angewählt werden können. Man kann den Inhalt einer Diskette ansehen, ein altes oder neues Dokument laden oder einen DOS-Befehl geben. Farbeinstellungen, Laufwerkparameter und Druckerparameter werden erst viel später eingestellt, nämlich dann wenn man sie braucht. Wer nichts ändern möchte, wird aber auch nicht dazu gezwungen. Eine sinnvolle Eigenschaft von Vizawrite 64 ist es, alle wesentlichen Parameter wie Farbgebung, Zeilenformat, Druckertyp und Druckerparameter mit dem Dokument abzuspeichern. Der Anwender merkt davon nichts: Er erhält ein einmal abgespeichertes Dokument wieder so auf den Bildschirm, wie er es abgespeichert hat, ohne sich Gedanken über Formulare machen zu müssen. Damit ist allerdings der Nachteil verbunden, daß es bei Vizawrite 64 gegenüber Textomat Plus etwas länger dauert, das gesamte Format eines Textes zu ändern.

Textomat Plus arbeitet gegenüber Vizawrite 64 nur teilweise bildschirmorientiert. Das heißt, der Text erscheint im Schreibmodus nur annähernd in seiner späteren Form.

Zwar besitzt auch Textomat Plus nun einen sogenannten »Word wrap«, der ein Wort, das nicht mehr ganz in eine Zeile passen würde, mit in die nächste Zeile übernimmt. Prinzipiell faßt Textomat Plus den Text aber als eine fortlaufende Folge von Zeichen auf, die erst durch entsprechende Formatbefehle im Text ihr späteres Aussehen erhalten. Damit sind einige Nachteile verbunden, denn der beschriebene Text kann, besonders wenn öfters korrigiert oder ergänzt wurde, recht unübersichtlich werden. Das gilt wohlgemerkt nur im Schreibmodus, in der Funktion »Ansehen«, die den Text mit 40- und sogar 80 Zeichen auf den Bildschirm darstellt, werden überzählige Leerstellen ignoriert und der Text, besonders im 80-Zeichenmodus, so gezeigt, wie er später tatsächlich aussieht. Leider sind in dieser Funktion keine Korrekturen am Text möglich. Man muß sich entweder alle Fehler, die behoben werden sollen, aufschreiben, oder öfters zwischen Schreib- und Ansehmodus wechseln. Bei längeren Texten ist das aber ein vollkommen unpraktisches und sehr zeitraubendes Unterfangen, denn um von einem Modus in den anderen zu gelangen, muß Textomat Plus immer wieder auf die Masterdiskette zugreifen (Ladedauer im 80-Zeichenmodus insgesamt 48 Sekunden). Eine leichte Abhilfe ist es, wenn man das File »80« auf die Textdiskette kopiert, denn dann spart man sich wenigstens den Diskettenwechsel. Etwas unverständlich ist bei diesem Konzept die Funktion der variablen Zeilenbreite. Sie ist im Normalfall auf 40 Zeichen eingestellt, das heißt nachdem der rechte Bildschirmrand erreicht wird springt der Cursor in eine neue Zeile. Wer möchte, kann aber auch bis zu 240 Zeichen in eine Zeile eingeben, der Text wird dann horizontal verschoben. Eine solche Funktion ist aber eigentlich nur bei einem rein bildschirmorientierten Textsystem sinnvoll, in der Regel wird man deshalb auch bei 40 Zeichen pro Zeile bleiben.

## Vizawrite 64 kann's besser

Vizawrite 64 ist dagegen vollkommmen bildschirmorientiert. Der Text wird im Schreibmodus so eingegeben und formatiert, wie er später aussehen soll. Der Text wird dabei, entsprechend der in der jeweiligen Formatzeile angegebenen Zeilenbreite, horizontal verschoben. Alle Absätze, Einrückungen und Zentrierungen haben direkte Wirkung auf das Aussehen des Textes. Vizawrite 64 besitzt keinen »Anseh-Modus«, mit dem der Text in seinem späteren Aussehen, möglicherweise sogar in 80-Zeichendarstellung, betrachtet werden kann. Wer den Text als ganzes sehen möchte, ohne dabei mit dem Cursor durch die Zeilen fahren zu müssen, kann den gesamten Text mit der »Width-«Funktion blitzschnell auf 40 Zeichen umformatieren. Da in diesem Modus weiterhin geschrieben werden kann, ist die Korrektur eines Textes schnell durchführbar. Aber auch beim Editieren eines Textes ist Vizawrite 64 seinem Konkurrenten überlegen. Bei Vizawrite 64 können beliebige Textteile (auch mitten aus einer Zeile heraus) kopiert, dupliziert oder aber gelöscht werden. Die betreffenden Textteile werden durch überfahren mit dem Cursor (auch Zeilen- und Seitenweise möglich) hell unterlegt. Durch Bestätigung mit der RETURN-Taste wird die gewünschte Funktion mit sehr hoher Geschwindigkeit ausgeführt. Die Bedienung ist dabei denkbar einfach. Soll beispielsweise mitten in einer Zeile ein ganzer Satz eingefügt werden, genügt ein Druck auf die F7-Taste, um beliebig viele Zeichen einzufügen. Drückt man nochmals auf die F7-Taste, wird der restliche Text perfekt an den neu geschriebenen Text angefügt.

Mit Textomat Plus ist das schon etwas komplizierter. Zunächst gestattet dieser nur das zeilenweise Kopieren oder Verschieben, wobei zunächst ein Block markiert werden muß. Zum anderen ist ein Löschen nur zeilenweise, oder ab Cursor/bis Cursor möglich. Ein Einfügen von Text ist ähnlich wie bei Vizawrite 64, leider werden dabei immer wieder Leerzeichen hinzugemogelt, die zwar nicht den Ausdruck verändern, wohl aber die Übersichtlichkeit erschweren. Diese Leerzeichen werden deshalb eingefügt, weil Textomat Plus bei allen Editierbefehlen immer nur zwei Zeilen hinter dem Cursor erkennt. Beim Löschen entsteht deshalb in der zweiten Zeile eine Lücke. Übrigens ist es bei Textomat Plus nur im Einfüge-Modus möglich, einen nachträglichen Absatz einzufügen, denn sonst wird beim Drücken von SHIFT-RETURN der Rest der Zeile einfach gelöscht. In Praxi sind somit im ungünstigen Fall bis zu vier Tasten notwendig, um einen Absatz nachträglich zu setzen. Auch sollte die Funktion von RETURN (Zeilenwechsel ohne Absatz) und SHIFT-RETURN (Zeilenwechsel mit Absatz) vertauscht werden.

Textomat Plus arbeitet mit einer Vielzahl von Menüs (Bild 1). Diese




**Bild 2. Eine interessante Möglichkeit des Textomat Plus: Text und Grafik mischen.**

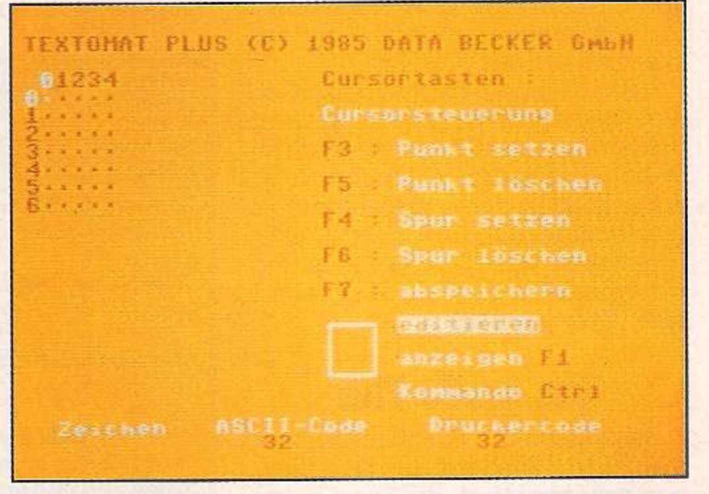



**Bild 3. Mit dem eingebauten Hilfsprogramm lassen sich beim Textomat Plus sehr einfach neue Zeichen definieren**

Menüs zum Ansteuern der Befehle sind alle hierarchisch geordnet. Um zu einer bestimmten Funktion zu kommen, bietet Textomat Plus zwei Wege an. Entweder wird von oben nach unten durch alle Menüs gewandert, oder aber der direkte Weg durch Drücken einer bestimmten Taste im Menü-Modus gewählt. Leider ist letzterer Weg zurück zum Schreibmodus nicht möglich. Es bleibt oft nichts anderes übrig, als immer wieder auf die Funktionstaste F2 zu drücken, bis man wieder im Schreibmodus ist. Eine »Escape«-Taste, die sofort wieder in den Schreibmodus führt, wäre sicherlich eine sinnvolle Ergänzung.

Vizawrite 64 kennt dagegen nur zwei Menüs: Das Haupt- und das Druckermenü. Alle wesentlichen Funktionen werden direkt durch bestimmte Tasten in Verbindung mit der Commodore-Taste aufgerufen. Ein effizientes Verfahren, an das man sich schnell gewöhnt.

Bislang haben wir hauptsächlich die Standardfunktionen unserer beiden Testkandidaten betrachtet. Dabei ist es natürlich unmöglich, alle Funktionen direkt miteinander zu vergleichen. Eine Übersicht aller Funktionen von Textomat Plus und Vizawrite 64 ist deshalb in Tabelle 1 zusammengestellt.

## Das große Plus

Beim Textomat Plus wurden erstmalig eine Reihe von Sonderfunktionen programmiert, die sonst in keinem anderen Textverarbeitungsprogramm für den Standard C 64 zu finden sind. In diesem Abschnitt soll darauf etwas genauer eingegangen werden.

Textomat Plus verarbeitet Grafiken! Einerseits ist es möglich Grafiken einzuladen, zu betrachten und ganz oder abschnittweise abzuspeichern. Andererseits werden durch ein spezielles Steuerzeichen Grafiken, die im Standardformat ab dezimal 8192=hex 2000 abgespeichert sind, im Text eingefügt (Bild 2). Leider kann bei dieser Funktion der Grafikabschnitt immer nur vertikal und nicht horizontal bestimmt werden, deshalb ist es auch nicht möglich, links oder rechts neben eine Grafik etwas zu schreiben. Das geht nur, wenn man das Papier entweder von Hand zurückdreht oder bei Druckern mit Rückwärtstransport die entsprechenden Steuerzeichen sendet.

Jedes Terminalprogramm ist eigentlich auch ein kleines Textverarbeitungsprogramm. Warum sollte deshalb nicht in einem Textverarbeitungsprogramm auch ein Terminalmodus vorhanden sein? Textomat Plus besitzt eine spezielle Funktion zum Senden und Empfangen von Texten via Akustikkoppler. Am günstigsten ist es, wenn empfangene Texte von einem Computer gesendet wurden, der ebenfalls mit Textomat Plus arbeitet. In diesem Fall kommt es zu keinerlei Problemen beim Datenempfang. Soll aber beispielsweise mit einer Mailbox kommuniziert werden, wird es notwendig, zuerst eine RS232-Parametertabelle und dann eine Liste der ASCII-Zeichen einzugeben. Hier wäre es wünschenswert, wenn die gängigsten Normen auf der Masterdiskette mitgeliefert würden oder wenigstens im Handbuch aufgeführt wären. Mit einem speziellen Terminalprogramm ist Textomat Plus allerdings nicht zu vergleichen.

### Allerlei Druckmodi

Das wohl interessanteste am Textomat Plus sind seine verschiedenen Druckmodi. Er besitzt drei davon, mit jeweils ganz unterschiedlichen Fähigkeiten. Der wohl gebräuchlichste ist der Standarddruck, bei ihm werden die Zeichen in Form eines Codes an den Drucker geschickt. Das hat den Vorteil, daß der Ausdruck relativ schnell vonstatten geht und alle Steuerzeichen (Fettschrift, Unterstreichen) verarbeitet werden.

Im Grafikdruck werden alle Zeichen über Grafik erzeugt (auch auf den Commodore-Druckern!). Das bedeutet, daß eine selbst entworfene Schrift und diese sogar in proportionalem Zeichenabstand auf das Papier gebracht werden kann. Diese Methode verlangsamt den gesamten Druckvorgang natürlich wesentlich, außerdem werden keine

| Leistungsmerkmal \ Textverarbeitungsprogramm | Vizawrite 64 | Textomat Plus |
|---|---|---|
| Kopieren von Textteilen | Ja, beliebig viele Zeichen | Ja, aber nur zeilenweise |
| Disk-Befehl senden | Ja, im Hauptmenü und im Schreibmodus | Ja, im Menü-Modus |
| Finden eines Textausdrucks | Ja, mit und ohne Berücksichtigung von Groß- und Kleinschreibung | Ja, im Menü-Modus |
| Ersetzen eines Textausdrucks | Ja, entweder überall oder nur beim ersten gefundenen Ausdruck | Ja, entweder überall oder nur beim ersten gefundenen Ausdruck |
| Bestimmte Seite anwählen | Ja | Nein |
| Verschieben von Textteilen | Ja, beliebig viele Zeichen | Ja, aber nur zeilenweise |
| Merge — Anhängen von Texten | Ja, von einem Laufwerk | Ja, von zwei Laufwerken |
| Neubenennung eines Textes | Ja | Ja |
| Speichern auf Kassette/Diskette | K/D | D |
| Farbwahl | Ja | Ja |
| Schreiben auf Bildschirmbreite | Ja | Ja |
| Word-wraps | Ja | Ja, abschaltbar |
| Einfügen von Text | Ja, beliebig viele Zeichen | Ja, beliebig viele Zeichen |
| Löschen von Text | Ja, beliebig viele Zeichen | Ja, aber nur zeilenweise, einzelne Buchstaben, ab Cursor, bis Cursor |
| Anzeige freier Speicherplatz | Ja | Nein |
| Tabulatoren | Numerisch/Text | Text |
| Zentrieren | Ja | Ja |
| Eliteschrift | Ja | Ja |
| Einrücken | Ja | Ja |
| Breitschrift | Ja | Ja |
| Unterstreichen | Ja | Ja |
| Doppeldruck | Ja | Ja |
| Sub/Superscript | Ja | Ja |
| Fettdruck | Ja | Ja |
| Seitennumerierung | Ja | Ja |
| Eigene Druckercodes | Ja, unbegrenzt viele frei definierbar | Ja, zehn frei definierbar |
| Blocksatz | Ja, im ganzen Text (wahlweise) | Ja, an- und abschaltbar |
| Einzelblatt-STOP | Ja | Ja |
| Mit Text belegbare Tasten | Nein | Ja, 8 verschiedene |
| Leerzeilen per Steuerzeichen | Nein | Ja |
| Adressen einfügen | Ja, von verschiedenen Programmen, vorzugsweise Vizastar 64 | Ja, von verschiedenen Programmen |

Steuerzeichen mehr akzeptiert. Einzige Ausnahme sind Drucker mit einem ladbaren Zeichensatz. Wer sich auf eine 11*8 Punktmatrix beschränkt, kann auch weiterhin Steuerzeichen verwenden.

Als Mischdruck wird beim Textomat Plus ein Druckverfahren bezeichnet, das sich aus dem Standarddruck und dem Grafikdruck zusammensetzt. Dieser Mischdruck erlaubt die Definition eigener Sonderzeichen, die dann im Grafikdruck eingefügt werden.

**Hilfsprogramme eingebaut**

Natürlich ist es ohne geeignete Hilfsprogramme nicht gerade einfach, einen neuen Drucker- oder Bildschirmzeichensatz zu entwerfen. Deshalb hat man den Textomat Plus auch mit einer ganzen Reihe von Programmierhilfen ausgestattet. Ein komfortables Programm (von T. Müller) das direkt aus dem Textmodus geladen werden kann, erfüllt alle Ansprüche an einen komfortablen Bildschirm- und Zeicheneditor (Bild 3). Die Definition neuer Zeichen wird so kinderleicht. Besondere Befehle erlauben es, ein Zeichen vielgestaltig zu bearbeiten. Wer mit neuen Schriften, aber auch mit wissenschaftlichen Sonderzeichen arbeiten muß, wird von dieser Funktion des Textomat begeistert sein.

**Flexibilität großgeschrieben**

Textomat Plus ist so programmiert, daß es mit möglichst vielen Druckern zusammenarbeitet. Neben der bereits vom Vorgänger her bekannten Tabelle zur ASCII-Code-Zuweisung, sind bereits auf der Masterdiskette einige Steuer-Codes wichtiger Druckertypen abgespeichert (siehe Tabelle). Textomat Plus arbeitet außerdem mit einigen Datenbanken (Superbase, Datamat) zusammen, um Serienbriefe erstellen zu können. Für kleinere Anwendungen ist es sogar vorgesehen, eigene Adreßdateien anzulegen, die später im Text eingefügt werden. Endlich ist es auch möglich, den Druck vorzeitig abzubrechen.

Textomat Plus ist ohne Zweifel eines der besten Textverarbeitungsprogramme für den C 64. In einigen Bereichen ist er sogar unerreicht. Leider weist der Schreib-Modus einige Mängel auf, die Textomat Plus von der professionellen Ebene (in der er sich preislich bewegt) trennen. Die exzessive Menütechnik ist Geschmacksache. Bei längerem Arbeiten wird es allerdings lästig, die Menüs rauf und runter zu wandern, man tut gut daran die Befehlstasten schnell zu lernen. Der größte Mangel des Textomat Plus ist aber die extrem unkomfortable Art der Textkorrektur. Es kann niemanden zugemutet werden, sich alle Änderungen, die ihm im »Anseh-Modus« auffallen, zu merken, um sie später im Schreibmodus zu korrigieren. Vorbildlich ist dagegen die Hilfsprogrammsammlung. Es gibt kein anderes Textverarbeitungsprogramm, das auch nur ähnliche Manipulationen des Bildschirm- und Druckerzeichensatzes zuläßt.

Wenn es darum geht, schnell und sehr komfortabel Texte zu erstellen, ohne dabei auf spezielle Grafikschriften zurückzugreifen, gibt es nach wie vor kaum ein besseres Programm als Vizawrite 64. Die reichhaltige Ausstattung und die Möglichkeit, beliebig viele Druckbefehle zu definieren, heben Vizawrite 64 eindeutig in die professionelle Klasse. Es ist natürlich Ansichtssache, aber es scheint, daß ein konsequent bildschirmorientiertes Textsystem am vorteilhaftesten ist. Unerreicht ist bislang die Korrektur- und Editierfunktion von Vizawrite 64. Schon oft haben wir das beliebige Umkopieren oder Löschen einzelner Textteile schätzen gelernt. Mit 34 000 freien Zeichen ist der Textspeicher von Vizawrite 64 übrigens um 10 000 Zeichen größer als beim Textomat Plus.

Wer gerne mit Schriften experimentiert, oder aber Grafik und Text mischen möchte, aber auch diejenigen, die oft spezielle Sonderzeichen benötigen, sind mit Textomat Plus (248 Mark) bestens bedient. Für alle, die schnell, komfortabel und vor allem übersichtlich Texte erstellen möchten, bleibt Vizawrite 64 (298 Mark) die beste Empfehlung.

(Arnd Wängler/gk)

| Leistungsmerkmal / Textverarbeitungsprogramm | Vizawrite 64 | Textomat Plus |
|---|---|---|
| Grafik einfügen | Nein | Ja |
| Datum autom. einfügen | Nein | Ja |
| Rechnen im Text | Nein | Ja |
| Trennvorschlag vorgeben | Nein | Ja |
| Modem-Modus | Nein | Ja |
| Grafik einlesen/ausdrucken | Nein | Ja |
| Text zeigen | Ja, im 40-Zeichen-Modus mit Editierfunktion | Ja, im 40- oder 80-Zeichen-Modus ohne Editierfunktion |
| Rundschreiben | Ja | Ja |
| Bildschirmzeichensatz ändern/anzeigen | Nein | Ja |
| Druckerzeichensatz ändern/anzeigen | Nein | Ja |
| ASCII-Zuweisungstabelle | Nein | Ja |
| Umlaute auf CBM-Drucker | Nein | Ja |
| Proportionalschrift auf CBM-Druckern | Nein | Ja |
| Anzahl der Laufwerke ändern | Ja | Nein |
| Zeichensatz laden | Nein | Ja |
| Tastaturbelegung | deutsch | ASCII/deutsch |
| Handbuch (Note) | 3 | 2 |
| Textausgabe auf Disk | Nein | Ja |
| Abspeichern auf Kassette | Ja | Nein |
| Besonders unterstützte Druckertypen | ASCII-Drucker<br>VC1525/1526/MPS802/<br>MPS801/MPS803<br>Epson MX/RX/FX 80/100<br>Qume Diabolo<br>Spinwriter<br>Triumph Adler TRD 170S<br>Ricoh Flawriter<br>Brother CE50/51/60/70<br>Juki 6100<br>Petal MA 20 | ASCII-Drucker<br>VC1525/1526/MPS802/<br>MPS801/MPS803<br>Epson MX/RX/FX 80/100<br>CP80X<br>Panasonic<br>Quendata<br>Quelle Privileg<br>Smith Corona<br>Brother HR15 |
| Centronics-Schnittstelle eingebaut | Ja | Nein |
| Druck über RS232-Schnittstelle | Nein | Ja |
| Preis | 298 Mark (Disk)<br>340 Mark (Modul) | 248 Mark (Disk) |

**Tabelle 1. Textomat Plus und Vizawrite 64 im Vergleich.**

Bezugsquellen: Textomat Plus: Data Becker, Postfach 1440, 4000 Düsseldorf 1, Vizawrite 64: Interface Age, Josephsburgstr. 6, 8000 München 80


64ER ONLINE

# Macro-Basic: Die Unterprogramm-Bibliothek

**Herkömmliche Basic-Erweiterungen sind in der Regel aus einem Guß, mit vielen nützlichen, aber auch unnötigen Befehlen. Und oft fehlen gerade die Funktionen, die man am dringendsten braucht. Damit will Macro-Basic ein für allemal Schluß machen.**

An anderen Befehlserweiterungen fehlt immer etwas: Exbasic Level II hat keine Grafikmöglichkeiten, Simons Basic kein richtiges RENUMBER, um nur einige Beispiele zu nennen. Der Gedanke, ein flexibles Software-Instrument zu bauen, welches aus einer Vielzahl von Modulen besteht, die man — je nach aktuellem Bedarf — in den Computer lädt, drängt sich da geradezu auf. Verwirklicht wurde dieses einleuchtende Konzept mit Macro-Basic.

Packt man Macro-Basic aus, dann fällt dem in dieser Hinsicht von Commodore nicht gerade verwöhntem C 64-Besitzer zunächst einmal ein mehrere hundert DIN-A4-Seiten starkes Handbuch auf. Da merken Sie dann, was es heißt, seinen verfügbaren Basic-Befehlssatz in etwa zu vervierfachen. Das ist aber noch nicht alles, denn im Mai soll noch eine Grafikerweiterung auf den Markt kommen mit nochmal 18 neuen Befehlen. Daneben übersieht man fast die doppelseitig beschriebene Diskette und den Dongle, einen Kopierschutzstecker, der in den Kassettenport geschoben werden muß.

Wie ist Macro-Basic aufgebaut? Zwei Softwarepakete bilden derzeit das komplette System, von denen das erste aus einem Ladeprogramm und 260 Befehlen besteht, das zweite aber eine Dateiverwaltungs-Bibliothek mit weiteren 30 Befehlen bereithält. Das Ganze verteilt sich auf 160, größtenteils voneinander unabhängige, Module, die mittels eines Linkprogramms nach Belieben zusammengestellt werden können. Theoretisch verfügt man mit Macro-Basic nicht nur über eine, sondern über 160 x 160 = 25600 mögliche Befehlserweiterungen, je nachdem, wie man sich die Module zusammenbastelt. Jedes Modul enthält einen oder meist mehrere Befehle in Form von optimierten Maschinenprogrammen. Das kleinste Modul beansprucht 7 Byte, das größte — wenn ich mich nicht vertan habe — 3670 Byte an Speicherplatz. Im Schnitt opfert man für jedes Modul etwa 100 bis 200 Byte. Wohin man im Speicher seine Befehlsbibliothek packen will, bleibt im eigenen Ermessen. Eine fertig erstellte Basic-Erweiterung kann unter einem Kennwort abgespeichert werden. Sollten Sie Zugang zu anderen Commodore-Computern haben, dann können Sie die Befehlserweiterung auch auf diese anwenden, denn Macro-Basic ist mit besonderem Augenmerk auf Kompatibilität gebaut worden.

Das Erstellen der individuellen Basic-Erweiterung geschieht relativ einfach im Dialog. Dazu lädt man das erste Programm auf der Macro-Basic-Diskette, gibt ein, auf welchem Commodore-System man arbeitet, steckt auf Anforderung den Datenstecker in den Kassettenport und gibt an, wo im Speicher die Maschinenprogramme abgelegt werden sollen. Man hat dann die Möglichkeit, sich auf dem Drucker ein Protokoll der weiteren Prozedur mitschreiben zu lassen. Das empfiehlt sich besonders bei größeren Erweiterungen, weil man nach einiger Zeit sicher nicht mehr weiß, welche Kombinationen man gewählt hat, und wohin man im Speicher das ganze Werk lädt. Die Startadresse braucht man nämlich, um die Erweiterung zu initialisieren. Von nun an ist alles ein Kinderspiel. Das Programm erfragt Stück für Stück die Kennzahlen der einzelnen Module bis man schließlich END eingibt. Dann erteilt man dem Ganzen einen Namen, legt eine formatierte Diskette ins Laufwerk, der Drucker schließt das Protokoll ab und das Ladeprogramm speichert alles ab, bevor es sich diskret aus dem Geschehen zurückzieht. Zum Testen habe ich mir mal eine Erweiterung aus 154 Modulen zusammengestellt, die etwa 23,6 KByte beansprucht. Das Ganze dauert ungefähr eine halbe Stunde. Wenn man mehr Übung darin hat, reduziert sich der Zeitaufwand sicher noch erheblich. In diesen 23,6 KByte liegen immerhin 260 neue Befehle! Für den Normalgebrauch kommt man sicher leicht mit den 4 KByte im oberen RAM des Commodore 64 ($C000 bis $D000) aus.

Benötigt man nun irgendwann eines der auf diese Weise zusammengestellten Instrumente, dann lädt man es unter dem zugeteilten Namen, gibt danach NEW ein und startet mittels SYS Startadresse,* . Die Startadresse holt man sich aus dem Protokoll als Startadresse Original. Automatisch wird nun auch der Basic-Speicher begrenzt und alle Befehle der Erweiterung stehen zur Verfügung.

Um welche Befehle es sich dabei handelt, sehen Sie aus der beigefügten Tabelle. Es ist allerdings unmöglich, in diesem Rahmen alle 290 Befehle vorzustellen. Deswegen habe ich nur eine — sicherlich etwas willkürliche — Auswahl der wichtigsten dort aufgeführt. Allen Macro-Basic-Befehlen ist ein »&« voranzustellen.

Sind Sie ein Verfechter der strukturierten Programmierung? Dann verfügen Sie über Befehle wie IF... THEN... ELSE... ENDIF, aber auch SELECT.. CASE.. OTHER.. ENDSEL, DO...DO WHILE...DO UNTIL... mit END DO und EXIT. Label können gesetzt und mit GOSUB oder GOTO angesprungen werden. Echte Unterprogramme mit eigenen Variablen, die per CALL anzusteuern sind, sind möglich. Variable und Arrays können aber auch vom Hauptprogramm an das Unterprogramm übergeben werden. Mittels ' (Hochkomma) können Kommentare ins Programm geschrieben werden. Durch SUBMIT ist es möglich, auf einfache Weise mit Overlay-Technik zu arbeiten, was auch durch den OVERLAY-Befehl für einzelne Programme möglich ist. ON ERROR und RESUME erlauben das Auffangen von Fehlermeldungen, die in Variablen EC%, EL% und ET$ als Fehlercode, fehlerhafte Basic-Zeile und Fehlertext abfragbar sind. Fast schon selbstverständlich sind Befehle wie AUTO (AUTONUMBER-

Funktion, die auch noch Texte wie zum Beispiel DATA schreiben kann), BLOAD oder BSAVE (ladender oder speichernder Zugriff auf beliebige Speicherbereiche), DELETE (Löschen beliebiger Zeilen), DLINK (Anhängen von Programmen) und DMERGE (Mischen von Programmen nach Zeilennummern!), DUMP-Funktionen, die auch die Auflistung von Arrays erlauben, LIM (zum Begrenzen des Basic-Speichers), LPAGE (zum seitenweisen Vorwärts- oder Rückwärts-Auflisten von Programmen), RENUM (das ist ein echtes RENUMBER), RLOAD (zum Laden und Starten eines Programmes), FIND und SEARCH (durchsucht Programme nach Variablen, Befehlen oder Konstanten und listet nach Speicherposition oder Basic-Zeilennummer auf) und TRACE-Möglichkeiten zur schrittweisen Bearbeitung von Basic-Programmen.

Der Verkehr mit Peripherie wird gefördert durch SPOOLING-Befehle, Anweisungen zur Druckeranpassung, DISK zum Ermitteln von Diskettenparametern, HCOPY zum Ausdrucken des Bildschirminhaltes. SCR LOAD oder SAVE erlaubt das Einladen oder Abspeichern von Bildschirmen. Mittels RLIST und SLIST lassen sich relative oder sequentielle Dateien auf dem Bildschirm zeigen. TURBO beschleunigt Diskettenoperationen.

Grafiken können — ab Mai angekündigt, auch mit Hochauflösung — durch Pixel oder Balken erstellt werden. FLASH-Funktionen lassen Bildschirmteile blinken, SOUND und BEEP sorgen für Lautuntermalungen. Eine Reihe von Befehlen greifen direkt auf den Bildschirm zu und erlauben beispielsweise das Füllen von Fenstern mit Zeichen.

Mathematisch Interessierte haben eine Vielzahl von neuen Möglichkeiten zur Verfügung, angefangen vom ACOS über Rechnungen mit mehrfach genauer Zahlendarstellung (bis zu 100 Ziffern!), Umrechnungen innerhalb der Zahlensysteme dezimal/sedezimal/binär, Winkelmaß-Umrechnungen etc. bis zur Berechnung des Gregorianischen Datums aus dem Julianischen und umgekehrt. Strings können wie mathematische Ausdrücke behandelt oder sogar als Basic-Anweisungen interpretiert werden.

Und beim Stichwort Strings sind wir bei der eigentlichen Stärke von Macro-Basic angelangt. Ich habe nie gewußt, was man alles mit Strings anstellen kann, bis ich Macro-Basic kennenlernte! Sie werden kaum eine String-Operation

## Auswahl von Macro-Basic Befehlen

| Befehl | Kurzbeschreibung |
|---|---|
| AUTO | Automatische Zeilennumerierung mit Textvorgabe-Option |
| BIN | Wandelt Binärstring in Dezimalzahl um |
| BIN$ | Wandelt Ganzzahl in 8- oder 16-stelligen Binärstring um |
| BLOAD | Datenblock ab bestimmter Adresse in Speicher laden |
| BSAVE | Speicherbereich abspeichern |
| BSPACE | Bildschirmfenster mit Zeichen füllen |
| CALL | Aufruf echter Unterprogramme |
| CEIL | Ermittelt den Nachkommawert |
| CHANGE | Durchsucht Speicher nach Stringvariable, teilt Position mit und ersetzt String |
| CONV | Umsetzen von Druckersteuerzeichen etc. in einem String |
| DEEK | Lesen von 2 Byte, die in 16-Bit-Wert umgesetzt werden |
| DEG | Umrechnung Altgrad in Bogenmaß |
| DELDIM$ | Löschen eines String-Arrays |
| DEZ | Umwandeln von Sedezimalzahlen in Dezimalzahlen |
| DLIST | Anzeige best. Records auf dem Bildschirm |
| DMERGE | Mischen von Basic-Programmen nach Zeilennummern |
| DOKE | Speichern von 16-Bit-Werten als 2 Byte |
| EDIT | Editieren bestimmter Bildschirmzeilen |
| EVAL | Ermittelt Zahlenwert aus Arithmetikstring |
| FAK | Berechnen der Fakultät |
| FIND | Durchsucht Speicher nach String und teilt Position mit |
| FLASH | Läßt Bildschirmfenster blinken |
| FREDIM | Hebt eine Dimensionierung auf |
| GET | Liest Zeichen von Tastatur und überträgt nur erlaubte Zeichen |
| INDEX | Durchsucht Stringvariable nach einer Stringvariablen und teilt Position mit |
| INFORM | Erlaubt formatierte Dateneingabe |
| INMSG | Überträgt Speicherbereich in Eingabefelder einer Bildschirm-Maske |
| INPUT # | Lädt maximal 65000 Zeichen von Diskette in angegebenen Speicherbereich |
| INS$ | Einfügen eines Strings in eine Stringvariable |
| KEY.S$ | Durchsucht sequentielle Datei nach Suchbegriff |
| LIKE | Kopiert Speicherbereich in Stringvariable |
| LINE INPUT | Überträgt alle Zeichen (auch ,; etc.) in Stringvariable |
| LIST$ | Überträgt Basic-Zeile entschlüsselt in String |
| MASK | Erlaubt die Erstellung von Bildschirmmasken |
| MAX | Ermittelt Maximum aus numerischen Variablen |
| MOB DEF | Erlaubt die temporäre Anwendung weiterer Erweiterungen von Diskette |
| MIN | Ermittelt Minimum aus numerischen Variablen |
| MOVE | Kopiert Speicherbereich an vorgegebene Adresse |
| NONSTOP | Blockiert STOP-Taste |
| NSORT | Sortiert Gleitkomma-Array in aufsteigender Folge |
| PACK | Packt Gleitkommavariable in 5-Byte-String |
| PLIST | Zeigt Basic-Programm von Diskette auf Bildschirm |
| QUIT | Liest nur erlaubte Zeichen von Tastatur |
| REP | Schaltet Repeat-Funktion ein |
| REPLC$ | Durchsucht String A$ nach String B$ und ersetzt B$ durch C$ |
| RESTORE | Setzt DATA-Zeiger auf bestimmte Zeile oder Label |
| ROLLDE | Löscht Bildschirmzeile. Überträgt die Zeile in String und füllt sie dann mit einem angegebenen Stringinhalt |
| ROTL | Rollt Stringinhalt nach Bedingungen nach links |
| ROUND | Rundet Zahl auf angegebene Nachkommastellen |
| RUN$ | Ausführen von Basic-Befehlen in einem String |
| SBC | Subtrahiert Inhalt zweier Stringvariablen |
| SCR DO | Rollt Bildschirm nach unten |
| SET% | Setzt Bitposition eines Strings auf bestimmten Wert |
| SIM | Simuliert Eingabe einer Stringvariablen über Tastatur |
| SORT | Sortiert String-Array in aufsteigender Folge |
| STATE | Setzt Inhalt einer String-Variablen in ausführbare Basic-Befehle um und plaziert sie in angegebene Basic-Zeile während der Ausführung |
| SWAP | Vertauschen zweier Zahlenvariable |
| TEST% | Ermittelt Bit-Wert bestimmter Position eines Strings |
| TFRAME | Erzeugt Rahmen um Textstring |
| TIME | Initialisiert, stellt und zeigt Uhr an bestimmter Speicheradresse |
| TRANS | Formatiertes Umsetzen eines Strings nach Format und Positionsvariable |
| TRON | Aktivieren der TRACE-Funktion |
| USING | Ausgabe von Zahlen oder Strings nach Formatvariablen |
| WAIT STOP | Hält Programm an, bis STOP-Taste ein zweites Mal gedrückt wird |
| XRT | Ermittelt X-te Wurzel aus Y |
| ZADR | Ermittelt Adresse des 1.Byte einer bestimmten oder der gerade ausgeführten Basic-Zeile. |
| , | Erlaubt Kommentar auch zwischen Befehlen. Vor jeden Befehl & setzen. |

vermissen. Mir ist jedenfalls keine mehr eingefallen. Ob es sich um Suchen von ganzen Strings oder Teilen davon, Tauschen oder Verändern, Ein- oder Ausgabe, Positionieren, Prüfen, Umwandeln, Formatieren, Löschen... dreht, alles ist vorhanden.

Stringverarbeitung spielt beim Verwalten von Daten eine Rolle. Neben einer ganzen Reihe von neuen Befehlen, die den Datenzugriff auf Diskette oder im Speicher betreffen, gibt es sechs Module mit 30 Befehlen, die einen Dateityp schaffen, der ISAM/VSAM genannt wird. ISAM kennt vielleicht manch einer als IBM-Begriff: Indexed Sequential Access Method (also Index-sequentieller Dateizugriff). VSAM steht für Virtual Storage Access Method, also virtuelle Dateizugriffsmethode. Mittels ISAM/VSAM kann auf dreierlei Wegen Dateiverarbeitung geschehen: Sequentielle Verarbeitung in der Eingabereihenfolge, sequentielle Verarbeitung in Schlüsselfolgen und die gezielte Verarbeitung über Schlüssel. Datensätze dürfen variable Länge haben, die Datenspeicherung erfolgt blocküberspannend, mehrere Schlüssel dürfen pro Datensatz verwendet werden, aber auch mehrere Datensätze pro Schlüssel sind möglich, wobei übrigens der Schlüssel nicht Bestandteil des Datensatzes sein muß. Bis zu fünf Diskettenstationen der unterschiedlichen CBM-Formate können parallel verwaltet werden. Außer den Befehlen zum Einrichten, Löschen und Sichern solcher Dateien stehen Befehle zum Verarbeiten von Diskettenfehlern, zum Update, zum Suchen von Datensätzen, zum Überschreiben, zum Kopieren, zum Reorganisieren der Datei beziehungsweise der Einträge zur Verfügung.

Sicherlich werden Sie in dieser Beschreibung Befehle vermissen. Das liegt aber nicht daran, daß Macro-Basic sie nicht zur Verfügung stellen würde, sondern daran, daß es unmöglich ist, hier alle aufzuführen. Kritik muß das Handbuch finden. Nicht, daß es vor Rechtschreibfehlern nur so strotzt stört, sondern, daß es offensichtlich für die großen Brüder des Commodore 64 geschrieben wurde, denn alle den Bildschirm betreffenden Operationen beziehen sich auf einen Bildschirmstart bei 32768. Einige Demoprogramme enthalten schlichtweg Fehler. Der Gebrauch der Bezeichnung BCD (binary coded decimal) als Abkürzung für Bildschirmcode verwirrt auch einigermaßen. Der Autor versichert aber, das Handbuch sei inzwischen überarbeitet worden und die neue Version enthalte diese Fehler nicht mehr. Der Preis hält sich im üblichen Rahmen: 298 Mark zahlt man für die Normalausführung (inklusive Dongle), dazu noch 34 Mark für das Handbuch. Als Zusatzpackung erhält man für 148 Mark die ISAM/VSAM Dateipackung. Nach Auskunft des Autors sind der Grafikzusatz und weitere geplante Ergänzungen darin schon enthalten.

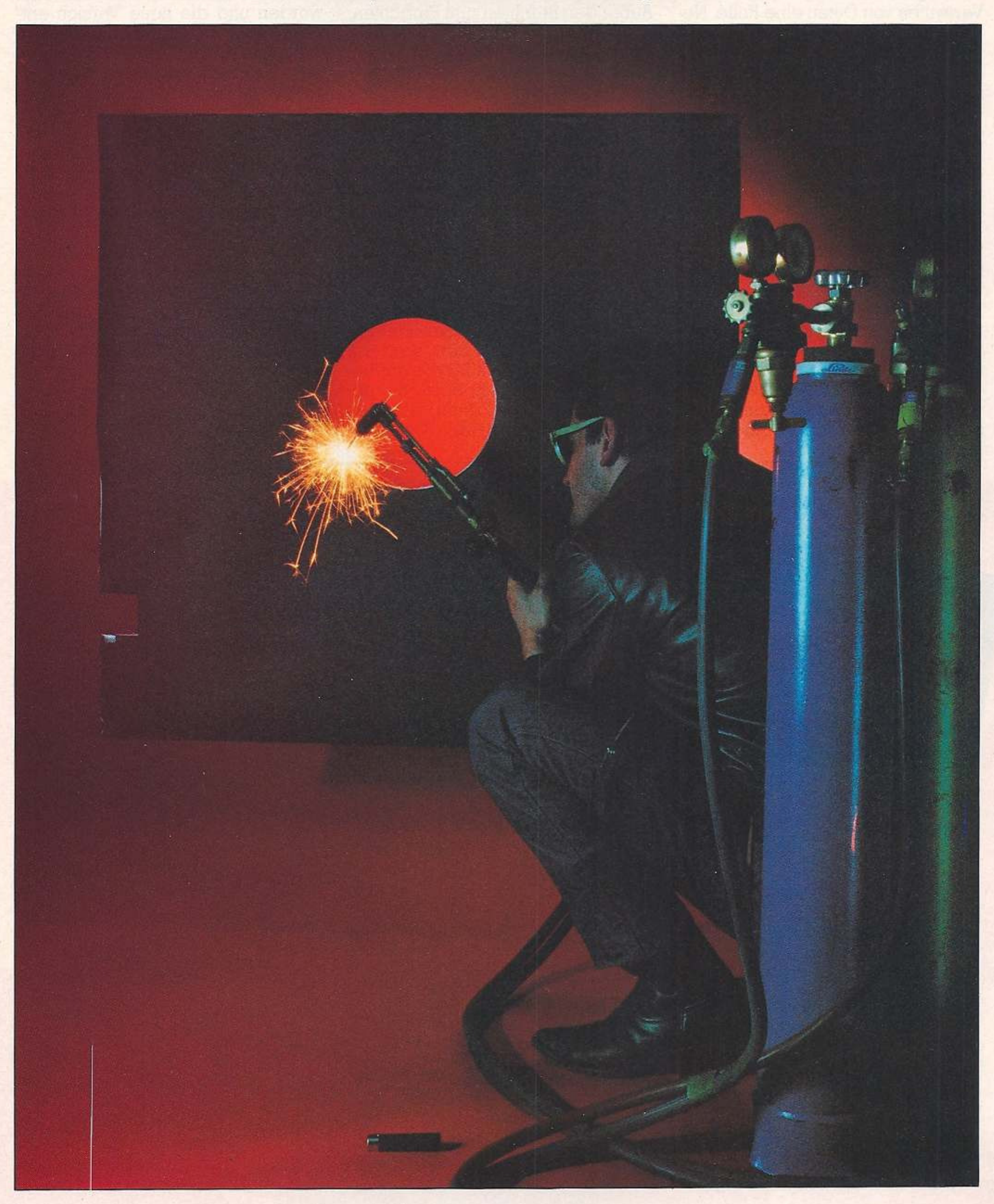
Fazit: Durch Macro-Basic wird der Commodore 64 zu einem derart vielseitigen Computer, daß dieses Software-Paket für ernsthafte Anwender guten Gewissens zu empfehlen ist. (Heimo Ponnath/gk)

Info: S&A Software, Bernd, Lambertusstr. 32, 5210 Troisdorf-Muellekoven



# Directory-Manipulationen

**Wer seine Programme vor den Augen anderer schützen möchte, der muß heutzutage schon zu recht harten Methoden greifen. Erstaunlich ist, was man allein schon mit dem Directory einer Diskette anfangen kann.**



So mancher benutzt den C 64 doch tatsächlich noch, um Programme selber zu schreiben, und nicht nur zur Anwendung von »vorgefertigter« Software, nicht zuletzt Spielen. Und diese selbstgeschriebenen Meisterwerke sollen vor fremdem Zugriff, also dem Kopieren geschützt werden. Allerdings haben sich gerade manche dieser unermüdlichen Freaks darauf spezialisiert, Kopierprogramme zu schreiben, die einfach alles kopieren, was sich auf der Diskette befindet, seien das »Errors«, »Halb-« oder »verschmierte Spuren«, »Lükken« und »Synchronmarkierungen«. Ein wirksamer Kopierschutz ist also kaum noch realisierbar. Eines kann man aber nach wie vor: Die Programme wenigstens so sicher machen, daß keiner einen Blick hinter die Kulissen werfen, oder gar Teile in eigene Programme übernehmen kann. Meistens ist dann von »Autostart« und »Listschutz« die Rede. Ein Kapitel, mit dem man sich noch relativ wenig beschäftigt hat, sind die Directories der Disketten. Gerade diese lassen sich mit relativ wenig Aufwand so geschickt »vermurksen«, daß ein nicht Eingeweihter beim Versuch an das Programm heranzukommen, dasteht, wie der Ochs vorm Berg.

Alle hier vorgestellten Methoden haben allerdings einen Nachteil: Sie schützen nicht vor Kopierversuchen mit Programmen, die ganze Disketten blockweise kopieren (FCopy, Quickcopy und ähnliche). Sie verhindern aber den Einblick in die abgespeicherten Programme, unter denen sich dann natürlich ein Schutzprogramm befinden kann. Alles, was Sie im folgenden benötigen, ist ein Diskettenmonitor, beispielsweise den SMON, sowie Grundkenntnisse über die Blockorganisation auf Diskette.

## Der Trick mit dem Gänsefüßchen

Ein absoluter Verblüffungseffekt läßt sich mit minimalem Aufwand erreichen, man benötigt noch nicht einmal einen Diskmonitor. Werfen Sie doch mal einen genauen Blick auf das Directory in Bild 1. Und nun überlegen Sie sich mal, wie Sie das zweite Programm laden können, das von der Blocklänge her das Hauptprogramm sein müßte. Das Problem scheint unlösbar, lädt doch LOAD "*",8 immer das erste Programm. Welchen Filenamen hat denn das zweite Programm überhaupt? Nun, des Rätsels Lösung lautet: Das Filename lautet » " «. Dies ist kein Druckfehler, sondern tatsächlich ein einzelnes Gänsefüßchen. Nun, selbst mit dieser Information scheinen wir aber immer noch nicht weiter zu kommen: LOAD""",8 erzeugt eine Fehlermeldung. Falls Sie selbst drauf kommen wollen: Denken Sie an den ASCII-Code. Und für alle anderen, die es nicht selber ausknobeln wollen: Der entsprechende LOAD-Befehl lautet: LOAD CHR$(34),8

Das Gänsefüßchen hat den ASCII-Code 34. Der einzige Weg, dies als Filenamen an die Floppy zu senden, ist die CHR$-Funktion. Das Ganze funktioniert analog dazu beim SAVE- und VERIFY-Befehl.

Und das ist noch nicht alles: Probieren Sie mal doch spaßeshalber folgenden SAVE-Befehl aus: SAVE "64"CHR$(34)"ER",8. Nachteil dieser Schreibweise: Das Programm läßt sich vom Unkundigen auch mit LOAD"64*",8 laden. Wichtig zu wissen ist nur folgendes: Tauchen im Filenamen ein oder mehrere Gänsefüßchen auf, so folgt kein abschließendes Gänsefüßchen. Der ganze Dreh beruht auf einigen Macken in der Directory-Sende-Routine des Floppy-DOS.

Noch eines ist interessant: Hinter dem Anführungszeichen im Filenamen werden die folgenden Steuercodes (Farbwechsel, Cursorbewegungen und ähnliches), sofern deren ASCII-Codes unter 128 liegen, nicht in Grafikzeichen übersetzt, sondern ausgeführt. Farbige Directories? Warum nicht!

## Schräge Typen

Hiermit sind nicht die dunklen Gestalten aus Gangsterfilmen gemeint, sondern die vermurkste Anzeige des Filetyps im Directory. Schauen Sie noch einmal auf das Beispieldirectory in Bild 1. Glauben Sie mir, daß der dritte Eintrag kein sequentielles File, sondern ein Programm ist? Schön und gut, werden Sie sagen, aber wie lad ich's? Auch hier darf über die Einfachheit gestaunt werden: Es genügt ein einfaches LOAD"6510,S,R",8 und schon wird ganz brav das doch eigentlich sequentielle File in den Computerspeicher befördert. Natürlich muß hier vorher mit SAVE"6510,S,W" gearbeitet worden sein. Leider kann man auch echte sequentielle Files so laden, was dann und wann zu einem sauberen Systemabsturz führen kann. Damit die Vielfalt gewährt bleibt, funktioniert der Trick auch mit USR-Files, dann muß allerdings auch im LOAD-Befehl ein »U« anstelle eines »S« stehen.

Probleme gibt es allerdings bei relativen Files. Hier weigert sich die Floppy beharrlich, selbige als Programm an den Computer zu senden.

So, nun wird es etwas komplizierter. Als nächstes wollen wir einzelne Files kurzzeitig »wiederbeleben«. Dazu brauchen wir aber die Block-Lese- und Schreib-Befehle.

Im folgenden benötigen wir eine leere Diskette. Dann speichern wir das zu sichernde Programm ab und danach ein Ladeprogramm, über das gleich noch zu reden sein wird. Wichtig ist, daß diese Reihenfolge streng eingehalten wird. Und nun benutzen wir den Scratch-Befehl der Floppy, um das erste File, also das Hauptprogramm, wieder zu löschen. Warnung! Ab sofort dürfen keine Schreibzugriffe mehr auf diese Diskette ausgeführt werden. Im Directory ist jetzt also nur das zweite File, der Lader, verzeichnet? Weit gefehlt: Das erste File steht jetzt immer noch auf der Diskette und im Diskettendirectory, allerdings mit dem Filetyp *DEL. Der Stern vor dem DEL ist ebenfalls kein Druckfehler, sondern bedeutet in der Directory-Sprache, daß ein nicht ordnungsgemäß geschlossenes File vorliegt. Solche Sternchen findet man meist bei sequentiellen Dateien, wenn man einen Fehler gemacht hat, oder bei Programm-Files, wenn diese nicht mehr auf die Diskette paßten. Und da sich unsere Floppy intelligent schimpft, unterschlägt sie die *DEL-Einträge beim Laden von Directories in den Computerspeicher, um uns nicht mit nicht mehr vorhandenen Files nerven zu müssen. Natürlich ist das File auch noch auf der Diskette vorhanden. Um es wieder zu regenerieren, muß das Filetyp-Byte des Eintrages auf den Wert 130 ($82) gesetzt werden. Beim ersten Eintrag ist das Filetyp-Byte das Byte Nummer 2 (also das dritte! Zählungen beginnen bei Computern grundsätzlich bei Null) im Block Track 18, Sektor 1. Im Lader sollte nun in etwa folgende Befehlssequenz stehen:

```
OPEN 3,4,3,"#"
OPEN 15,8,15
PRINT#15,"U1:3 0 18 1"
PRINT#15,"B—P:3 2"
PRINT#3,CHR$(130)
PRINT#15,"U2:3 0 18 1"
PRINT#15,"I"
CLOSE 15,3
LOAD":*",8
```

```
0 [garbled directory header]
5     "LADEPROGRAMM"     PRG
147   ""                 PRG
23    "6510"             SEQ
664 BLOCKS FREE.

READY.
```

**So kann ein »vermurkstes« Directory aussehen**

Der letzte LOAD-Befehl lädt nun nicht, wie man bei einem Blick auf das Directory vermuten würde, das Ladeprogramm, sondern das Hauptprogramm, das wir jetzt regeneriert haben. Wichtig ist, daß die Diskette vor dem Laden des Hauptprogramms initialisiert wird. Die Floppy merkt sich nämlich immer einige Teile des Directories, und liest diese gar nicht erst von der Diskette, wenn ein Directory vom Computer abgerufen wird. Die Folge wäre, daß wir immer noch nicht mit dem alten Directory arbeiten, obwohl schon das neue auf der Diskette steht. Beim Initialisieren werden dann allerdings die internen Zwischenspeicher der Laufwerks aktualisiert.

Die ersten Befehle im Hauptprogramm sollten das Hauptprogramm auf der Diskette wieder »löschen«, indem die obere Sequenz verwendet wird, aber anstelle des CHR$(130) ein CHR$(0). Dies ist die Kennzeichnung für ein *DEL-File. Das geht schneller und ist weniger auffällig als ein Scratch-Befehl.

Dieses Prinzip läßt sich fast beliebig ausweiten und stark verbessern, so zum Beispiel wenn man die Filenamen verändert oder die Angabe der Anzahl der Blöcke, die ein Programm belegt. Wer hier ansetzen will, der sollte sich ruhig noch einmal die Folgen Eins und Zwei unseres Floppykurses ansehen (64'er, Ausgabe 10 und 11/84), dort sind alle wichtigen Informationen über den Directoryaufbau enthalten.

Ein Nachtrag zu Bild 1: Falls Sie die komische FREE-Meldung irritiert haben sollte: Sie läßt sich durch Freigeben aller Blöcke in der BAM mittels Diskmonitor erreichen.

## Ein Directory verschwindet ...

Eine gute, alte Lebensweisheit lautet: »Wenn du nicht überzeugen kannst, verwirre!« Ähnliches haben wir jetzt vor, denn jeder, der sich von den bisherigen Tricks nicht überzeugen ließ, den versuchen wir jetzt durch ein nicht vorhandenes Directory in tiefste Zweifel an der eigenen Intelligenz (und der der Floppy) zu stürzen.

Es gibt zwei verschiedene Methoden, Directories gegen Einblick zu sichern. Die bekanntere folgt zuerst: Wer sich ein wenig mit der internen Struktur des C 64 auskennt, der weiß, daß der LIST-Befehl nur so lange arbeitet, bis er auf die Bytefolge 0,0,0 also dreimal Null, stößt. Dann wird das LISTen eines Programms sofort abgebrochen. Nun, was ist ein Directory, sofern es im Computerspeicher steht, aber anderes als ein nicht lauffähiges aber LISTbares Programm? Bringen wir irgendwo die drei Nuller unter, haben wir schon fast gewonnen. Wo man die drei Nuller versteckt, hängt ganz vom Benutzer ab: Ganz zu Anfang im Diskettennamen oder erst später in irgendeinem Filenamen, so daß man zumindest noch die ersten Filenamen lesen kann.

Dieser Trick in Verbindung mit anderen (Cursorbewegungen, Reversumschaltungen), sorgt schon für einigermaßen große Verblüffungseffekte.

Nachteil dieser Methode: Sie ist schon sehr bekannt, und außerdem steht das Directory im RAM des Computers und kann entweder mit einem Monitor oder aber mit einigen Tricks wieder LISTbar gemacht werden.

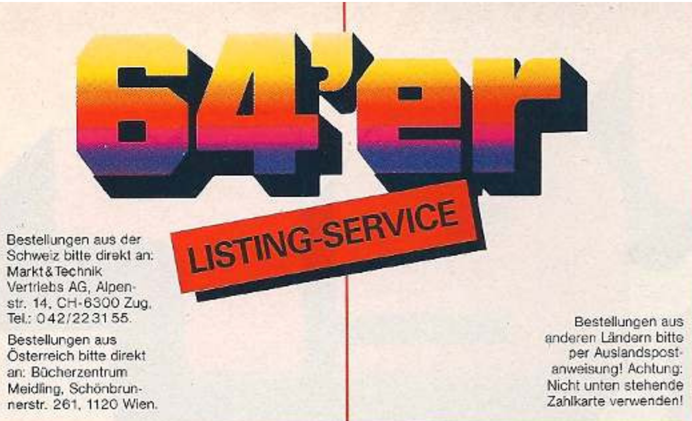
Noch gemeiner und sicherer, aber einfacher, kann man folgendermaßen vorgehen: Nehmen wir an, auf der Diskette befinden sich maximal acht Programme. Damit ist nur der Directoryblock 18,1 belegt. Die ersten beiden Bytes dieses Blocks enthalten die Werte 0 und 255, um 18,1 gleichzeitig als letzten Directoryblock zu kennzeichnen. Ändern Sie diese Werte mit einem Diskmonitor auf zwei völlig verrückte Werte, so zum Beispiel 64,85. Sollten Sie nun ein Directory laden wollen, beschwert sich die Floppy mit »ILLEGAL TRACK OR SECTOR«. Der Hammer bei der Geschichte ist aber: Es wird kein Filename an den Computer gesandt, beim Auflisten des Directories sieht man nur den Diskettennamen und die ID. Hier helfen dann auch keine Tricks und kein Monitor mehr: Das Directory, oder zumindest die vernünftigen ersten Einträge kommen nie beim Computer an.

Trotzdem funktionieren LOAD-Befehle immer noch einwandfrei, die floppyinternen Funktionen des Directory sind also nicht außer Kraft gesetzt. Einziger Nachteil der Methode ist, daß man sich auf acht Fileeinträge beschränken muß. Natürlich lassen sich alle bisher vorgestellten Methoden beliebig miteinander kombinieren, auch wenn das im Augenblick sinnlos erscheint. Wieso sollte man schon die Filetyp-Einträge manipulieren, wenn man doch niemals ein Directory sieht? Aber gerade die Kombination macht es dem Neugierigen besonders schwer, hinter die Geheimnisse des Directories zu kommen.

## Die beste Methode »Compiler Schutz«

Zum Schluß noch einige allgemeine Tips: Schutzprogramme, die in Basic geschrieben wurden, sollten Sie unbedingt compilieren. Jeder Compiler erstellt einen derartigen »Wurschtel«-Code, der fast unlesbar ist, so daß da selbst die ausgebufftesten Knack-Profis resignieren müssen. Wichtig ist aber, daß Befehle, die an die Floppy gesandt werden, im Programmtext verschlüsselt stehen sollten. Zum Beispiel, in dem sie mit dem CHR$-Befehl arbeiten und die einzusetzenden Zahlenwerte über einen Entschlüsselungsalgorithmus errechnet werden. Dann kann man dem Compilat noch nicht einmal über einen Monitor entnehmen, welche Befehle zur Floppy gesandt werden. Soweit zu den Manipulationen an der Directory. Der Einblick in unsere Disketten ist nun verwehrt. Verwenden Sie dann auch noch einen Compiler, bleibt auch das Programm ein Buch mit sieben Siegeln. Nun sollten Sie auch noch Autostart-Programme verwenden. Wie man ein Autostart-File erzeugt, das mit verblüffenden Eigenschaften ausgestattet ist, werde ich in einer der nächsten Ausgaben zeigen.

(Boris Schneider/rg)




Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG, Alpenstr. 14, CH-6300 Zug, Tel.: 042/22 31 56.

Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien.

Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

## Programme aus früheren Ausgaben

### Sonderheft: Abenteuerspiele

**Bestell-Nr.: L685 S2 — DM 34,90***

Adventure 2000, Zauberschloß, Quasimodo, Odyssee, Mario, Creater, Zeittunnel, World, Crantor, Listing 1-27

### Ausgabe 5/85

**Bestell-Nr. L 6 8505 A — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Checksum. Schnell | S. 54 |
| MSE Lader | S. 55 |
| MPS 802 | S. 31 |
| Format-System | S. 147 |
| VIC | S. 175 |
| 6510 I | S. 71 |
| Sternenhimmel (AdM) | S. 57 |
| Assemblerkurs | S. 144 |
| Direktory-Sorter | S. 77 |
| Trick.OBJ | S. 65 |
| 3D-Movie-Maker (LdM) | S. 65 |
| Modulator (Heft 4) | S. 155 |

**VC 20**

| Programm | Seite |
|---|---|
| Checksummer | S. 54 |
| Minigrafik | S. 69 |
| Longscreen | S. 83 |

**C 16**

| Programm | Seite |
|---|---|
| Help & Trace | S. 84 |

### Ausgabe 4/85

**Bestell-Nr. L 6 8504A — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Checksum. Schnell | S. 54 |
| MSE Lader | S. 58 |
| S-MON Teil 5 | S. 67 |
| Window 64 | S. 88 |
| XBASIC | S. 59 |
| ASS. Kurs Teil 8 | S. 138 |
| Hypra-Perfect | S. 76 |
| 11 neue Einzeiler | S. 153 |
| Print-List | S. 79 |
| Netzgrafik | S. 71 |
| ROT-DEMO | S. 73 |
| X.Y-DEMO 1 u. 2 | S. 73 |
| Ostereier | S. 51 |
| Befehlserweiterung | S. 23 |

**VC 20**

| Programm | Seite |
|---|---|
| Checksummer | S. 56 |
| Supergrafik III | S. 73 |
| Prg. Schutz | S. 83 |

### Ausgabe 3/85

**Bestell-Nr. L 6 8503A — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Checksu. Schnell | S. 70 |
| MSE Lader | S. 79 |
| Hires-3/Teil 2 | S. 136 |
| Intellisearch $C | S. 153 |
| Son of Destroyer | S. 42 |
| Ligatab (LdM) | S. 63 |
| Fl. Basicloader | S. 82 |
| Floppylister | S. 82 |
| E.Zeichnen 1 und 2 | S. 72 |
| Meisterschütze (AdM) | S. 53 |
| Saver | S. 89 |
| Screensave | S. 88 |
| Pic-Loader | S. 90 |

**VC 20**

| Programm | Seite |
|---|---|
| Checksummer | S. 70 |
| IRQ-Clock | S. 156 |
| Autostart | S. 159 |

### Ausgabe 2/85

**Bestell-Nr. L 6 8502A — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Checksummer 64 | S. 65 |
| MSE | S. 68 |
| SMON (Teil 4) | S. 72 |
| Grab des Pharao (LdM) | S. 56 |
| Basic-Lader | S. 91 |
| RAM-Floppy | S. 92 |
| Notlandung | S. 156 |
| Hires-3 | S. 123 |
| Benchmark-Test | S. 37 |
| Format.Stringkur | S. 148 |
| Versch. Test.USR | S. 140 |
| Cursorsteuerung | S. 86 |
| Rolling-Data | S. 88 |
| Super-Memory | S. 81 |

**VC 20**

| Programm | Seite |
|---|---|
| Checksummer VC 20 | S. 65 |
| Familienplanung (AdM) | S. 53 |
| Super 8-Steuerung | S. 70 |
| Q+Bert (3K) | S. 78 |
| Joypaint | S. 143 |
| Bitmapping | S. 145 |
| Multicolor-Demo | S. 146 |

### Ausgabe 1/85

**Bestell-Nr. L 6 8501A — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Checksummer 64 | S. 72 |
| Handballtrainer (AdM) | S. 53 |
| SMON (Teil 3) | S. 69 |
| Hi-Eddi (LdM) | S. 57 |
| Hypra-Load mal vier | S. 82 |
| Tips und Tricks | S. 87 |
| Provic 64 | S. 76 |
| Eingabe (UPB) | S. 156 |

**VC 20**

| Programm | Seite |
|---|---|
| Checksummer VC 20 | S. 72 |

### Ausgabe 12/84

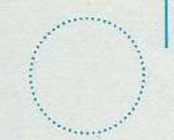
**Bestell-Nr. CB 022 — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Synthesizer (AdM) | S. 51 |
| SMON (2. Teil) | S. 60 |
| 3D-Vier gewinnt | S. 96 |
| Trace | S. 76 |
| Stringy | S. 88 |
| Lader | S. 92 |
| Auto | S. 84 |
| Listschutz | S. 85 |
| Simons Axo (SB) | S. 64 |
| Kreuzworträtsel | S. 150 |

**VC 20**

| Programm | Seite |
|---|---|
| Mathematikal Basic (8K>) (LdM) | S. 55 |
| Fast Tape | S. 80 |



### Ausgabe 11/84

**Bestell-Nr. CB 020 — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Turtle Grafik (LdM) | S. 48 |
| Schachmeister (AdM) | S. 50 |
| SMON (1. Teil) | S. 59 |
| Floppykurs | S. 117 |
| FPLOT-Befehlserweiterung | S. 73 |
| Get Koala pic | S. 66 |
| Interrupttechnik | S. 84 |
| Exsort (UPB) | S. 154 |
| Einzeiler | S. 158 |
| Simons Basic Befehlserweiterung (SB) | S. 90 |

**VC 20**

| Programm | Seite |
|---|---|
| Pseudosprites (8K) | S. 76 |
| Laterna Magica (8K) | S. 68 |
| Betriebssystem-Erweiterung (24K>) | S. 88 |
| Supergrafik (GV) | S. 71 |
| VC 20-Kurs (GV>) | S. 126 |

### Ausgabe 10/84

**Bestell-Nr. CB 019 — DM 29,90***

**Commodore 64**

| Programm | Seite |
|---|---|
| Finanzmathematik (AdM) | S. 68 |
| Hypra-Load (LdM) | S. 67 |
| Hardcopy Compact 2 | S. 86 |
| Hardcopy MPS 801 | S. 82 |
| Hardcopy VC 1526 neu | S. 83 |
| Hardcopy Gemini-10X | S. 85 |
| Hardcopy FX-80 | S. 88 |
| Hardcopy VC 1520 farbig | S. 84 |
| Apocalypse now | S. 106 |
| Supercopy | S. 102 |
| Disk-Dump | S. 95 |
| Diskettenorganisation | S. 97 |
| User-Port-Tastatur | S. 92 |
| Maske-(UPB) | S. 172 |

**VC 20**

| Programm | Seite |
|---|---|
| Epedemic | S. 112 |
| Video-Vorspann | S. 81 |

### Ausgabe 9/84

**Bestell-Nr. CB 014 — DM 29,90***

### Ausgabe 8/84

**Bestell-Nr. CB 013 — DM 29,90***

### Ausgabe 7/84

**Bestell-Nr. CB 017 — DM 29,90***

### Ausgabe 6/84

**Commodore 64**
**Bestell-Nr. CB 018 — DM 29,90***

**VC 20**
**Bestell-Nr. VC 008 — DM 29,90***

**Bedeutung der Abkürzungen**

- *LdM = Listing des Monats
- *AdM = Anwendung des Monats
- *SB = Simons Basic
- *GV = Grundversion
- *GV> = alle Speicherversionen können verwendet werden (einschließlich GV)
- *3K = 3-KByte-Speichererweiterung wird benötigt
- *8K> = Speichererweiterung größer als 8 KByte wird benötigt
- *UPB = Unterprogrammbibliothek

* Alle Preise inklusive Mehrwertsteuer.

**Fehlende Hefte erhalten Sie bei: Markt & Technik Vertrieb 64'er Hans-Pinsel-Str. 2, 8013 Haar**

---

**Empfängerabschnitt**

Postgirokonto Nr. des Absenders

DM | Pf

für Postgirokonto Nr.
**14 199-803**

**Lieferanschrift** und **Absender** der Zahlkarte

PLZ Ort

Verwendungszweck
**M&T Buchverlag Listing-Service**

**Zahlkarte/Postüberweisung**
für maschinelle Beschriftung

DM | Pf | für Postgirokonto Nr. **14 199-803**

Absender der Zahlkarte

PGiroA — Postgirokonto Nr. des Absenders — Postgiroteilnehmer

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postgirokontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung siehe Rückseite)

DM | Pf

für **Markt&Technik** Verlag Aktiengesellschaft
in **8013 Haar**

Postgirokonto Nr.
**14 199-803**

Postgiroamt
**München**

Ausstellungsdatum — Unterschrift

Für Vermerke des Absenders

Postgirokonto Nr. des Absenders

**Einlieferungsschein/Lastschriftzettel**

DM | Pf

für Postgirokonto Nr. **14 199-803** — Postgiroamt **München**

für **Markt&Technik** Verlag Aktiengesellschaft
**Hans-Pinsel-Str. 2**
in **8013 Haar**

Postvermerk

# New York City — The Big Apple & Air Support

Lange hat es ja gedauert, aber jetzt ist es da, oder besser gesagt, sind sie da, zwei neue Spiele der amerikanischen Softwarefirma Synapse. Beide Spiele bekommt man auf einer Diskette. Wie man es von Synapse gewohnt ist, zeichnen sich beide Spiele durch interessante Grafik und außergewöhnliche Spielideen aus. In »New York City — The Big Apple« wird der Spieler in die Lage eines Touristen versetzt, der es sich zur Aufgabe gemacht hat, möglichst alle Sehenswürdigkeiten der Großstadt zu besichtigen. Allerdings ist das für den armen Mann schwerer, als er sich das vorgestellt hat. Als Autofahrer muß er darauf achten, nicht in einen der vielen Verkehrsunfälle (fast jede Sekunde ein Unfall; die New Yorker sind rücksichtslose Autofahrer) verwickelt zu werden, als Fußgänger droht ihm der Tod durch Überfahren oder ein längerer, kostspieliger Krankenhaus-Aufenthalt. Nachdem er sein Auto sicher geparkt hat, beginnt ein Wettlauf mit der Zeit, da das Auto nach Ablauf der Parkuhr offiziell abgeschleppt wird. In diesem Fall steht ihm ein längerer Fußmarsch zur Sammelgarage bevor, wo er sein Auto gegen Entgelt wiederbekommt. Der Computer zeigt während des Spiels den Stadtplan von New York, und scrollt je nach Bewegung des Spielers in eine der vier Richtungen. Wird ein Gebäude besucht, wie zum Beispiel das Empire State Building, das World Trade Center oder der Zoo, so wird auf eine andere Spielszene umgeschaltet, in der der Tourist mit verschiedenen Gefahren zu kämpfen hat. So muß er kurzerhand im Post Office Briefe abstempeln und sortieren, im Zoo ausgebrochene Tiere einfangen, hat beim Geldabheben auf der Bank mit Bankräubern kämpfen oder läuft Gefahr, in der U-Bahn-Station erschossen zu werden. Insgesamt müssen zwölf Sehenswürdigkeiten besichtigt werden, bevor die aufregende Reise beendet ist.

## Air Support — Taktik und Strategie mit faszinierender Animation

Air Support, ein Taktik- und Strategiespiel, das auf der Rückseite der Diskette aufgezeichnet ist, weist eine faszinierende Animation auf. Es ist wirklich zu bewundern, wie Hunderte von Objekten fast stufenlos über den Bildschirm bewegt werden. Ziel des Spiels ist, mit einer Armee von Robotern gegnerische Roboter zu vernichten. Der Spieler steuert einen Helikopter über eine Landschaft aus Bäumen, Felsen und Wiesen, die je nach Bewegung nach oben oder unten gescrollt wird. Je nach Schwierigkeitsgrad wird er von feindlichen Geschützen beschossen und kann selber Bomben abwerfen. Es besteht die Möglichkeit, eigene Roboter aufzunehmen und mit dem Helikopter im Feindgebiet abzusetzen. Mit Hilfe von Sammelkommandos können ganze Armeen von über 100 Robotern über das Spielfeld gelenkt und zu Sperrlinien und Schlachtordnungen aufgestellt werden. Besonders, wenn man hohe Feindroboterzahlen wählt, wird der kleinste taktische Fehler zum Verhängnis. Air Support ist eine der interessantesten und abwechslungsreichsten Strategiespiele und erreicht auf alle Fälle die Qualität anderer Taktikspiele, wie zum Beispiel Combat Leader oder Operation Whirlwind.

Alles in allem hat Synapse es mal wieder geschafft, zwei hervorragende Spiele zu veröffentlichen. Der Preis von zirka 80 Mark für beide Programme ist daher durchaus akzeptabel. (H. J. Deeg/rg)

Info: Ariolasoft, Steinhauser Str. 3, 8000 München 80, Preis: etwa 80 Mark

# Des Rätsels Lösung

## Das Abenteuerspiel Amazon, erst in unserer vorletzten Ausgabe vorgestellt, ist schon gelöst.

Trotz der Lösung dieses Adventures wird Ihnen der Spaß nicht verdorben werden. Denn Amazon besitzt drei Schwierigkeitsgrade, von denen die letzten zwei noch zu meistern sind. Die Lösung zur Anfängerstufe stellen wir Ihnen hier vor.

Sie sollten aber nicht einfach abtippen. Das würde Ihnen den Spaß verderben. Benutzen Sie unsere Schritt-für-Schritt-Lösung am besten als Spickzettel, den Sie nur zwischendurch mal ansehen — aber auch nur dann, wenn Sie mal eine harte Nuß überhaupt nicht knacken können.

Und bevor Sie das tun, sollten Sie die Lösung lieber mit Papier abdecken und die folgenden kurzen Tips beachten:

Obwohl Paco ein sehr guter Helfer ist, sagt er nicht immer die Wahrheit.

Die »Goggles«, also die Brille, ist übrigens eine Infrarot-Brille. Sie ist gut bei Dunkelheit, besonders wenn man ganz bestimmte Leute nicht wecken will...

An der Brücke genügt es, Paco durch den Anblick des Gewehres zu überzeugen.

Vor dem Abflug nach Südamerika unbedingt den Spielstand abspeichern; Sie könnten die später so dringend benötigten Zigaretten womöglich schon am Flughafen loswerden.

Bei der Statue sollten Sie die Nordwand untersuchen. Und, was noch sehr wichtig ist: Die Bilder sind so aufgebaut, daß man immer nach Norden sieht. (M. Kohlen/rg)

WAIT, dann mit Joystick Sender einstellen, GO TO THE OFFICE, dann Name eingeben, gestellte Frage beantworten. Dann NOVICE eingeben. GO TO THE AIRPORT, OPEN LETTER, GO TO GATE 9, bei Taxifahrer: GO TO THE INSTITUTE, auf Frage (Hotel) mit NO antworten, ENTER INSTITUTE, SHOW LETTER, ENTER INSTITUTE. Geben Sie Ihren Namen ein und beantworten Sie die Frage mit YES, GO TO THE OFFICE, ENTER OFFICE, REMO-

VE DRAPE, Name eingeben, OPEN CAGE, OPEN REFRIGERATOR, GET MEDICINE, GET FRUIT, OFFER FRUIT, vier Fragen mit Yes oder Okay beantworten, PACO, OPEN CUPBOARD, INSPECT CUPBOARD, GET FOOD, INSPECT GLIMMER, GET GUN, INSPECT GUN, EXIT OFFICE, BRIBE DIRECTOR, dreimal NO sagen, bis er 5000 Dollar annimmt, GO TO THE AIRPORT, BUY TWO TICKETS, GO TO THE BOOTH, (Frage) YES, GET PACK, INVENTORY, Untersuchen Sie jetzt alles, um die Funktionen zu erfahren. Jetzt SAVE.

ENTER PLANE. Wenn jetzt jemand kommt, der Zigaretten haben will, dann LOAD. Weiter mit: SIT UP, WEST, WEST, ACTIVATE COMPUTER (Code=969), WEST, NORTH, NORTH, NORTH, WEST, SMILE, OFFER CIGARETTES, auf Frage (Food essen) mit YES antworten. NORTH, NORTH, N, N, WAIT, WAIT, auf die Frage irgend etwas antworten, nächste Frage mit YES beantworten, LOOK IN PACK, PACO, GET PACK, WEST, WEAR GOGGLES, N, REMOVE GOGGLES, ACTIVATE COMPUTER (diesmal den emergency code benutzen = 072), dem Computer die Richtung angeben, bis man die genannte Position erreicht (mit Hilfe der mitgelieferten Landkarte), NORTH, N, WEAR PARKA, N, N, ACTIVATE COMPUTER (wieder so wie oben verfahren), N, N, ENTER TUBE, OFFER FOOD, ACTIVATE COMPUTER (mit 969), SLEEP, N, N, N, N, E, E, E, E, GET PARACHUTE, OPEN CARTON, GET OAR, N, WEAR GOGGLES, FREEZE, W, W, REPAIR BOAT, USE PARACHUTE, SHOOT PACO, ENTER BOAT, CROSS RIVER, HIT ALLIGATORS WITH OAR, WAIT, ACTIVATE COMPUTER (mit 969), INVENTORY, ACTIVATE COMPUTER (969 etc., dann wieder Richtung eingeben, bis angegebene Stelle erreicht ist, dort im Actionspiel den Affen einfangen), E, E, S, S, S, W, ACTIVATE COMPUTER (969), SOUTH, ACTIVATE LASER AND WEAR GOGGLES (jetzt im Actionspiel die Huni-Warriors abschießen), SLEEP, N, GO IN, N, N, E, N, N, PUSH WALL, PULL LEVER, DOWN, W, UP, W, N, GET KEY, S, E, N, OPEN LOCK WITH KEY, GET PAPER, READ PAPER, ACTIVATE COMPUTER (969), W, OPEN 132. Jetzt dürfen Sie sich erstmal am Schatz und der Musik erfreuen, dann... alles bebt und: SOUTH, SOUTH, S, S, UP, WEST, ENTER THE HELICOPTER.

THE END

(M. Kohlen/rg)

# Interview

## Was sagt ein »Game Designer« über die Zukunft des Softwaremarktes?

David Crane, einer der Gründer von Activision und Autor von Softwarehits wie Pitfall, Decathlon und Ghostbusters stand uns Rede und Antwort. »Activision wird auch in der nahen Zukunft keine Anwendersoftware herausbringen; nur Unterhaltungssoftware. Wobei der Begriff Unterhaltungssoftware aber weiter ausgelegt werden muß, denn Unterhaltung besteht nicht nur aus Spielen. Activision wird in Zukunft auch neue Formen der Entertainment-Software bringen. Die ersten Schritte dazu machten wir schon mit Designer's Pencil und Music Studio.«

Die Zukunft der Computer innerhalb unserer Gesellschaft sieht Crane positiv: »Eine Welt ohne einen Computer in jedem Haushalt kann ich mir nicht vorstellen. Es wird zwar lange Dauern, aber die Entwicklung kann man mit dem Einzug des Fernsehers in das Wohnzimmer vergleichen. Der Hauptzweck des Computers ist eben die Unterhaltung; wäre dies nicht der Fall, würde ich die Entwicklung nicht so optimistisch beurteilen.« Wir befragten David Crane auch gleich über die Zukunftspläne von Activision im Zusammenhang mit den neuen Computern von Commodore und Atari. Cranes Antwort: »Wir werden für jeden Computer Software herstellen, der sich gut verkauft.« Unsere nächste Frage betraf speziell den PC 128. »Activision wird keine Spiele machen, die nur auf einzelnen Computertypen laufen. Spiele, die viel RAM benötigen, werden wir eher in mehrere Teile trennen (von Diskette nachladbar), damit jeder unsere Software genießen kann — auch Besitzer von kleineren Computern.« Und wie programmiert David Crane seine Spiele?

»Spiele entwickeln ist dem Drehen eines Filmes sehr ähnlich. Als erstes wird das Drehbuch geschrieben. Bei uns geschieht dies, indem der Game-Designer erst selbst Spiele spielt und mit ihnen herumprobiert. So lange, bis das Konzept (Drehbuch) fertig ist. Der Entwickler eines Spieles ist Drehbuchautor und Regisseur. Ein bißchen Kamera-Arbeit macht er auch. Die »Special Effects« wie Grafik oder Sound werden von Spezialisten gemacht.

Für ein Spiel benötigen wir in der Regel acht Monate bis ein Jahr. Dazu setzen sich Gruppen von etwa vier bis zwölf Programmierern zusammen, die gemeinsam ein Spiel erarbeiten, das von einem einzelnen Designer erdacht wurde.

Der Erfolg meiner Spiele liegt darin, daß ich immer voraussehen muß, was die Leute in einem Jahr wollen. Mittlerweile programmiere ich meine Spiele immer auf dem C 64, der zur Zeit mein Lieblingscomputer ist. Die Umsetzung auf andere Systeme wird dann von anderen Programmierern gemacht. Ghostbusters war übrigens mein C 64-Erstlingswerk.«

Wie bewertet David Crane den Einfluß des Deutschen Software-Marktes auf die USA? Viele Softwarehäuser in den USA scheinen den europäischen Markt gar nicht zu beachten. Oft müssen europäische Händler monatelang auf die Software aus den Vereinigten Staaten warten. Existiert der europäische Software-Markt für Amerika überhaupt? Wir bekamen auf diese Frage zwei Antworten von Crane. »Zum einen ist Europa ein sehr guter Stabilisierungsfaktor für uns. Läuft das Geschäft in den USA nicht, so läuft es in Europa. Die Konjunkturzyklen sind genau entgegengesetzt. Das andere ist etwas, was die Europäer noch gar nicht bemerkt haben. Amerikanische Marketing- und Vertriebsexperten behaupten, Europa sei dem US-Software-Markt um zwei Jahre voraus. In den USA wurde durch den Boom der Telespielbranche der Beginn des Heimcomputer-Zeitalters zurückgedrängt; in Europa wuchsen beide Sektoren zugleich. Sie waren gleichzeitig existierende Marktsegmente. Im Endeffekt hatte der europäische Software-Markt mehr Zeit zur Entwicklung als der amerikanische. Deshalb gibt es in den USA viele Fachleute, die sich das europäische Konsumentenverhalten ansehen, um die Entwicklung im eigenen Land besser voraussagen zu können.«

Europa, ein Testmarkt für das amerikanische Käuferverhalten der Zukunft? Nicht leicht vorstellbar, aber Insider wie David Crane scheinen davon überzeugt zu sein.

(M. Kohlen/rg)

# Semmelservice mit dem C 64

So ein Pech! Da hat man gerade eine Wette mit seiner Freundin verloren: Eine ganze Woche Frühstück ans Bett. Und das bei der Saukälte am Novembermorgen! Da fällt einem brühwarm der Handzettel ein, der vorgestern im Briefkasten lag. Jemand bot doch da einen Semmel- und Frühstücksdienst an? Der Blick in den Müll findet das gelbe Blättchen: Wie zu Omas Zeiten kann man sich Semmeln, Brezen, Milch, Butter und Eier morgens bis halb acht vor die Haustür bringen lassen. Die Laune steigt um hundert Prozent. Der Service rettet Ihre Morgenstunden.

Eines Tages hatte es der Medizinstudent Helmut Radspieler satt, »immer mit dem letzten Zwanziger in der Tasche« herumzulaufen. Da fiel ihm seine Jugend ein. Damals hatte er mit seinem Freund in einem amerikanischen Film (wo denn sonst?) gesehen, wie ein Botenjunge außer der Zeitung noch eine Tüte mit Brot und Butter an die Tür einer Vorstadtvilla schleuderte. Die damaligen Schüler versuchten die Idee in ihrer Wohngegend zu kopieren. Es funktionierte. Nur das Durchhaltevermögen der Jungs ließ nach, und nach dem Abi gaben sie endgültig auf. Bis vor zwei Jahren Helmut Radspieler das Geschäft wieder ankurbelte. Diesmal aber besser organisiert. »Professioneller« und mit Computerunterstützung. Als Computer benutzt er einen C 64 mit Diskettenstation und Drucker.



**Wie zu Omas Zeiten wird Ihnen in München das Frühstück vor die Haustür gebracht. Bei Sonnenschein und Regen. Der Unterschied zu früher: Anstelle eines Fahrrads ein Lieferwagen und ein C 64 für die Buchführung und Streckenplanung.**

## Freunde leisteten Hilfestellung

Ein Freund, Informatik-Student, hat Helmut dazu ein menügesteuertes Programm geschrieben, das drei Dateien verwaltet: eine Kundendatei, eine Bäckerliste und eine Tourenliste. Bäcker- und Tourenliste ergeben sich dabei jeweils auf Befehl aus der Kundendatei. Die Tourenliste wird nach den verschiedenen Gebieten aufgebaut, die vom Frühstückservice beliefert werden. Dort stehen in optimierter Reihenfolge die anzusteuernden Adressen. Dazu kommt bei jedem Kunden die Uhrzeit, zu der er spätestens seine Brötchen haben will. Ist der Fahrer mal in Verzug, muß er auch mal einen kleinen Umweg fahren, um einem Früh-aus-dem-Haus-Geher zu versorgen. Die Kundendatei enthält außer Namen und Adresse auch die Standard-Mengen, die ein Kunde pro Tag haben möchte. Will jemand nur am nächsten Dienstag drei Brötchen mehr, kann das genauso vermerkt werden, wie die Änderung »jeden Dienstag drei Brezen«. Per Telefon und Anrufbeantworter werden die vom Normalfall abweichenden Werte vom Kunden an Helmuts Frühstücksdienst und von diesem an den Bäcker und die Molkerei gemeldet.

## Ein typischer Arbeitstag:

»Um halb fünf morgens rasselt der Wecker, denn ich will mit Helmut auf Tour. In einer halben Stunde müssen wir in der Molkerei Schwabing die Milch abholen. Der weißbekittelte Helfer übergibt uns einen Korb mit 28 ganzen und 19 halben Litern Milch. Dazu elf Joghurt und 20 Halbpfünder Butter. Danach geht's zum Bäcker. Da liegen 687 Semmeln und Brezen, 15 Croissants und zwei Vollkornbrote für uns bereit. Wir laden das Zeug in einen japanischen Mini-Transporter ein. Damit grast »Helmuts Frühstücksservice« die Münchner Gegend zwischen Olympiazentrum und Schwabing ab. Ein angestellter zweiter Fahrer kutschiert die Ware durch Reihenhausgebiete im Münchner Westen. Den 150, meist festen Kunden kommt der Service nicht allzu teuer: Drei Mark die Woche, wenn jeden Tag geliefert wird und zwei, wenn's weniger als drei Tage sind. Auch die Preise für Semmeln, Croissants, Milch und Brezen sind kaum teurer als beim Bäcker oder dem Supermarkt an der Ecke. Die Masse macht's. Nachdem der zweite Fahrer bezahlt ist, bleibt ein hervorragender »Nebenverdienst«. Fast 3000 Mark im Monat bei dreieinhalb Stunden Arbeit am Tag, inklusive Buchführung.

Die Buchführung wird genauso wie das Schreiben von Rechnungen auch vom C 64 übernommen. Säumige Zahler werden gemahnt und Kunden-individuelle Fakturen geschrieben. Das Programm holt sich selbständig aus der Kundendatei die wöchentliche Bestellmenge und die Sonderlieferungen und Abbestellungen einzelner Tage. Die Preise pro Einheit werden getrennt ge-

Fortsetzung auf Seite 152

# Formatierte Eingabe

**Für professionelle Programme wird oft eine INPUT-Routine benötigt, die nur bestimmte Zeichen zuläßt. Mit diesem Maschinenprogramm können Sie sehr einfach Eingabemasken realisieren.**

Haben Sie schon mal wirklich professionelle Programme geschrieben? Dann hat Ihnen mit Sicherheit die INPUT-Routine des C 64-Betriebssystems zu schaffen gemacht, die sämtliche Zeichen zuläßt. Auch solche, die zu völligem Unsinn oder Fehlermeldungen führen. Dieser Mißstand wird in Basic-Programmen meist mit einer GET-Schleife umgangen. Bei dieser Methode wird ständig ein Zeichen von der Tastatur geholt, auf Zulässigkeit überprüft und auf den Bildschirm gedruckt. Die einzelnen Zeichen werden in einem String addiert. Wenn sie mit INS oder DEL im Eingabefeld Korrekturen ausführen wollen, können Sie sich vorstellen, wie kompliziert und zeitraubend eine solche Routine werden kann.

Eine schnelle Lösung dieses Problems schafft dieses Maschinenprogramm, das einen neuen Input-Befehl generiert. Die Syntax des Befehls lautet:

INPUT>Eingabelänge, Spalte, Zeile, Variable

Wie Sie schon an der Syntax sehen, ein recht komfortabler INPUT-Befehl. Mit der Eingabelänge geben Sie an, wie lang das Eingabefeld höchstens sein darf. Durch Angabe von Zeile und Spalte können Sie das Eingabefeld direkt an jeder beliebigen Bildschirmstelle positionieren. Als Variable müssen Sie eine Stringvariable einsetzen, zum Beispiel A$. Sollen ausschließlich Zahlen eingegeben werden, braucht der String nur noch auf Buchstaben hin überprüft werden. Beispielsweise mit folgender Routine:

```
10 INPUT>4,1,1,A$
20 IF STR$(VAL(A$))=" "+A$ THEN PRINT "ES IST EINE
ZAHL":END
30 PRINT "KEINE ZAHL":GOTO 10
```

**Tips zur Eingabe**

Das Programm müssen Sie mit dem MSE eingeben und abspeichern. Laden Sie dann die Input-Routine mit LOAD »name«,8,1 oder LOAD "name",1,1 und geben Sie NEW und SYS 49152 zur Initialisierung ein. Danach können Sie Ihre Basic-Programme eintippen und mit dem neuen INPUT>-Befehl verwenden. Die Routine können Sie natürlich auch mit dem Basic-Programm nachladen und mit dem neuen INPUT>-Befehl verwenden.

**Zusammenfassung**

— Die Routine wird mit SYS49152 initialisiert
— Das Programm benötigt den Speicherbereich von Adresse 49152 bis 49617
— Die Syntax lautet: INPUT>Eingabelänge, Zeile, Spalte, Stringvariable
— Die CLR-, HOME-, RUN-, STOP- und SHIFT-Taste ist bei der Eingabe gesperrt.
— Das Eingabefeld kann mit den Cursortasten nicht verlassen werden.
— INST/DEL wirkt nur innerhalb des Eingabefeldes
— Leerzeichen am Ende der Eingabe werden nicht mit übernommen.
— Benötigen Sie die Leerzeichen am Schluß geben Sie einfach POKE 49298,131 ein. POKE 49298,94 schaltet wieder in den Normalzustand.

Ein Nachteil soll nicht verschwiegen werden:

Fragetext kann nicht, wie beim serienmäßigen INPUT-Befehl, ausgegeben werden. Der Text muß mit PRINT gedruckt werden.

(Thomas Graf/hm)

```
PROGRAMM : E-ROUTINE 64   C000 C1F4
-----------------------------------

C000 : A9 0B A0 C0 8D 08 03 8C  AD
C008 : 09 03 60 20 73 00 C9 85  18
C010 : F0 06 20 79 00 4C E7 A7  8C
C018 : 20 73 00 C9 B1 F0 06 20  26
C020 : BF AB 4C AE A7 A9 00 8D  81
C028 : 6E C1 20 9B B7 8E 6F C1  23
C030 : 20 FD AE 20 9E B7 E0 28  7A
C038 : B0 0C 86 FD 20 FD AE 20  3D
C040 : 9E B7 E0 19 90 03 4C 48  F8
C048 : B2 8E 70 C1 A4 FD 20 0C  68
C050 : E5 20 24 EA A5 D2 85 FE  B1
C058 : 98 18 65 D1 90 02 E6 FE  43
C060 : 85 FD A5 F4 85 FC 98 18  BF
C068 : 65 F3 90 02 E6 FC 85 FB  90
C070 : A5 C6 85 CC 8D 92 02 F0  CB
C078 : F7 78 A5 CF F0 0C A5 CE  B2
C080 : AE 87 02 A0 00 84 CF 20  2A
C088 : 13 EA 20 B4 E5 C9 0D D0  31
C090 : 03 4C 71 C1 C9 14 F0 46  DB
C098 : C9 9D F0 42 C9 1D F0 29  50
C0A0 : C9 94 F0 61 C9 91 D0 03  8E
C0A8 : 4C 41 C1 C9 11 D0 03 4C  7B
C0B0 : 54 C1 AE 18 D0 E0 15 F0  DE
C0B8 : 08 C9 DB B0 B3 C9 C1 B0  A4
C0C0 : 08 C9 60 B0 AB C9 20 90  86
C0C8 : A7 C9 22 F0 A3 AE 6E C1  E7
C0D0 : EC 6F C1 F0 9B EE 6E C1  71
C0D8 : 20 16 E7 4C 70 C0 AC 6E  23
C0E0 : C1 F0 8D CE 6E C1 C9 9D  AE
C0E8 : F0 EE B1 FB AA B1 FD 88  7C
C0F0 : 91 FD 8A 91 FB C8 C8 F0  60
C0F8 : 07 CC 6F C1 90 EC F0 EA  83
C100 : A9 9D 4C D8 C0 AC 6F C1  59
C108 : CE 6F C1 88 C0 FF F0 2B  35
C110 : CC 6E C1 90 26 B1 FD C9  11
C118 : 20 F0 F0 CC 6F C1 F0 1B  85
C120 : B1 FB AA B1 FD C8 91 FD  18
C128 : 8A 91 FB 88 88 C0 FF F0  FB
C130 : 05 CC 6E C1 B0 EA C8 A9  48
C138 : 20 91 FD EE 6F C1 4C 70  95
C140 : C0 AA 38 AD 6E C1 E9 28  06
C148 : 90 07 8D 6E C1 8A 4C D8  E0
C150 : C0 4C 70 C0 AA 18 AD 6E  69
C158 : C1 69 28 B0 07 CD 6F C1  0E
C160 : F0 05 90 03 4C 70 C0 8D  BE
C168 : 6E C1 8A 4C D8 C0 EA EA  F8
C170 : EA 4C 96 C1 88 C0 FF D0  8E
C178 : 13 20 73 00 C9 00 F0 07  E7
C180 : C9 3A F0 03 4C 79 C1 A0  DC
C188 : 00 4C EB C1 B1 FD C9 20  54
C190 : F0 E2 C8 8C 6F C1 20 FD  36
C198 : AE 20 8B B0 48 98 48 20  FA
C1A0 : A3 B6 68 85 65 68 85 64  E2
C1A8 : AD 6F C1 20 75 B4 84 FB  88
C1B0 : A0 00 91 64 C8 8A 91 64  31
C1B8 : C8 A5 FB 91 64 A0 01 B1  37
C1C0 : 64 48 C8 B1 64 85 65 68  89
C1C8 : 85 64 AC 6F C1 88 B1 FD  BB
C1D0 : AE 18 D0 E0 15 F0 08 C9  67
C1D8 : 41 90 04 09 80 D0 06 C9  BE
C1E0 : 20 B0 02 09 40 91 64 C0  9E
C1E8 : 00 D0 E2 AE 70 C1 20 0C  8C
C1F0 : E5 4C AE A7 00 00 00 00  9C
```

**Listing zu »Formatierte Eingabe«. Das Listing ist mit dem MSE einzugeben.**




Die angegebenen Preise sind Ladenpreise.

# Wir suchen die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen VC 20 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren VC 20/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmerten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats

## 500 Mark

wert.

Schreiben Sie uns, was Sie mit Ihrem Computer machen:

Redaktion 64'er, Aktion: Anwendung des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

# Einmal im Monat gibt es die SUPERCHANCE

Diese nicht einmalige Gelegenheit sollten Sie nutzen. Wie? Schicken Sie uns Ihr bestes, selbst erstelltes Programm. Bei der Art des Programms sind wir nicht wählerisch.

Sie haben ein sehr gutes (Schieß-, Knobel-, Denk-, Action-, Abenteuer-)Spiel geschrieben: einschicken!

Sie verfügen über ein komfortables Disketten-Kopier-(Sortier) Programm mit einigen außergewöhnlichen Leistungsmerkmalen: einschicken!

Sie haben das Basic um einige sinnvolle Befehle erweitert: einschicken!

Sie arbeiten mit einem selbsterstellten Textverarbeitungsprogramm, einer eigenen Tabellenkalkulation, einem semiprofessionellen Datenverwaltungsprogramm: einschicken!

Sie zeichnen und konstruieren mit einem selbsterstellten Programm in hochauflösender Grafik: einschicken!

Wir freuen uns über jeden Beitrag und honorieren mit bis zu

## 2000 Mark

## für das Listing des Monats

Aus den besten Listings, die veröffentlicht werden, sucht die 64'er-Redaktion einmal im Monat das »Listing des Monats« aus. Alle Listings, die im 64'er abgedruckt sind, werden mit 100 bis 300 Mark honoriert. Die genaue Vorgehensweise beim Einsenden von Listings ist in dem Beitrag »Wie schicke ich meine Programme ein?« in verschiedenen Ausgaben beschrieben.

Schicken Sie Ihr Listing an: Redaktion 64'er, Superchance: Listing des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

Fortsetzung von Seite 147

halten, wie in jedem flexiblen System. Selbstverständlich werden Feiertage, an denen keine Semmeln ausgefahren werden, automatisch berücksichtigt. Das Kontrollieren der Rechnungen, die Helmut selber zahlen muß, also Molkerei und Bäcker, übernimmt ebenfalls der Computer. Man kann einen beliebigen Zeitraum eingeben, für den eine Rechnung erstellt oder überprüft werden soll. Mit dem Bäcker ergeben sich ab und zu Differenzen von vier oder fünf Brötchen, aber die regelt man problemlos. Der Bäcker kalkuliert und fakturiert ja auch per Computer.

Bei der morgendlichen Tour sehen wir keinen unserer Kunden. Die meisten von ihnen kennt auch Helmut nur per Telefon und Brief. Die Türen der Häuser, an denen er das Frühstück abliefert, sind um sechs Uhr früh geschlossen. Illustre Namen sind dabei: Freiherr von X, Graf Y und sogar Prinz Z. Dazu beliefern wir Münchens Villengegend. Da erstaunt es nicht, daß auch ein gewisser Herr Flick sich beliefern ließ. Junge Polizisten nehmen dankbar die Abwechslung in ihrer Dienstroutine zur Kenntnis.

Wir stellen die Papiertüten unter den Sims. Falls es regnen sollte, stopfen wir sie in bereitliegende Plastiktüten und schon hat uns die Straße wieder. Um halb acht sind wir fertig, das letzte Kipferl wartet auf den, der es verschlingen wird.

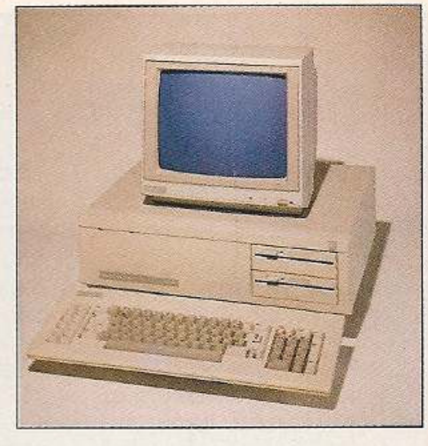
Der Frühsport — fahren, aussteigen, Autotür auf, Sachen raustragen, Tür zu, einsteigen, weiterfahren — steckt einem noch eine Zeitlang in den Knochen.

Leider kam an diesem Morgen keine Begegnung der Art zustande, wie sie mir Helmut schilderte: An einer Ampel bat ihn einmal eine junge Studentin um ein Brötchen. Er gab es ihr. Anschließend fuhr sie mit und half ihm bei der Arbeit. Als er schließlich das Angebot machte, ihr seinen C 64 zu zeigen, folgte sie ihm gerne zum Kaffeetrinken in seine Wohnung.

(Tomas F. Lansky/hm)

# Mitmachen — mitgewinnen

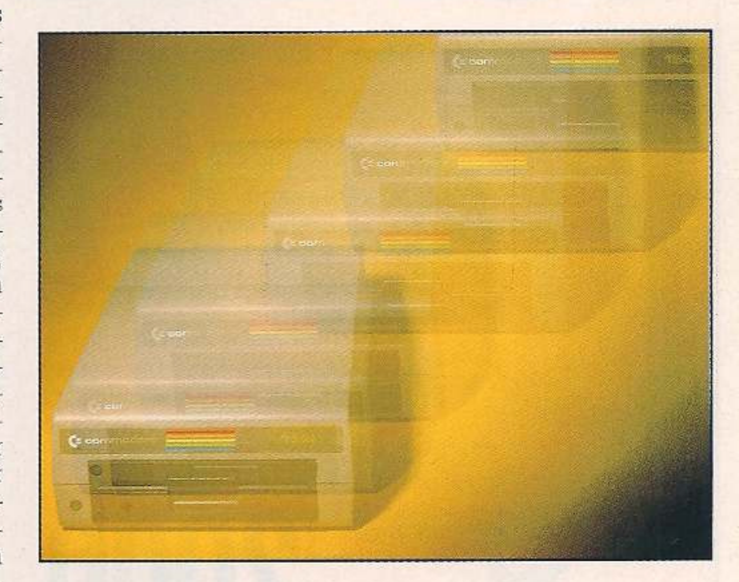
**Die 64'er will noch besser werden. Dafür müssen wir aber mehr über Sie, den Leser, wissen. Deshalb starten wir einen großen Umfragewettbewerb mit tollen Preisen.**



Was ist Ihre Aufgabe, um bei diesem Wettbewerb mitmachen zu können? Zunächst einmal, sollten Sie sich den Artikel über das Floppy-Laufwerk durchlesen, um die ersten sieben Fragen beantworten zu können. Nur wenn diese sieben Fragen richtig beantwortet sind, nehmen Sie an der Verlosung der Preise teil. Sodann bitten wir Sie den »Fragebogen« vollständig auszufüllen. Mit dieser Befragung haben Sie die Möglichkeit an der künftigen Gestaltung der 64'er mitzuwirken.

Der 64'er-Redaktion ist es bisher im Großen und Ganzen gelungen, ein in fast allen Belangen zufriedenstellendes Magazin für den C 64- und VC 20-Anwender zu machen. Doch der Markt ändert sich schnell, besonders auf dem Gebiet der Computerei. Nur wer genau weiß, was die Leser wollen, kann auch eine Zeitschrift machen, die für alle etwas beinhaltet. So haben wir beispielsweise erkennen müssen, daß nach mehr als einem Jahr seit dem die 64'er erscheint, wir uns von der Auswahl der Themen her immer mehr am Profi orientieren. Das hatte bittere Klagen der Anfänger und Fortgeschrittenen zur Folge. Eine andere Gruppe, die sich benachteiligt fühlte, waren die Datasette-Besitzer; ihr Argument: wir brächten zuviel für die Floppy. Auch die Basic-Programmierer fühlen sich benachteiligt, zuviel in Maschinensprache.

Sind das nun die Beschwerden von ein paar Unentwegten, oder steht eine schweigende Mehrheit dahinter? Ist Datenfernübertragung für alle interessant oder nur für Hacker? Begeistern Grafikprogramme oder Tabellenkalkulation mehr? Sollen wir mehr Listings für den technisch/naturwissenschaftlichen Bereich oder für die kaufmännische Anwendung bringen? Wie umfangreich sollen die Aktuellmeldungen sein oder bringen die Kurse mehr. Welche Kurse werden gewünscht? Ist die neue optische Aufmachung (Layout) bei Ihnen angekommen, oder sollen wir wieder mehr auf den »Bravo«-Stil umschwenken? Bevorzugen Sie die Möglichkeit das Gesicht »Ihrer« 64'er zu gestalten. Es ist klar, daß die Beantwortung der Fragen einige Zeit in Anspruch nimmt, und die 80 Pfennig sind auch kein Pappenstiel, doch wir glauben, es lohnt sich, vor allem für Sie. Je mehr mitmachen, umso sicherer wird die statistische Auswertung, desto besser wird die 64'er in der Zukunft. Und daß die 64'er eine Zukunft hat, daran glauben alle Redakteure und Mitarbeiter von Markt & Technik. Arbeiten Sie mit an Ihrer Zeitschrift.



Hardware- oder Software-Test? Was halten Sie von Bastelanleitungen? Soll der Computer-Markt ausgedehnt werden oder ist er gar zu groß? Welche Themen begeistern so viele, daß ein Sonderheft daraus gemacht werden kann?

Fragen über Fragen. Nur Sie als Leser können diese Fragen beantworten. Aber nur, wenn Sie sich so zahlreich wie nur irgend möglich beteiligen. Sie haben hier

Zum Thema Datenschutz: Ihre Adresse wird nicht zusammen mit den statistischen Daten, die lediglich für die zukünftige Gestaltung der 64'er von Interesse sind, abgespeichert. Sie dient lediglich dazu, Sie im Falle eines Gewinnes zu benachrichtigen. Schicken Sie den Fragebogen an Markt&Technik Verlag AG, Redaktion 64'er, **Stichwort: Umfrage,** Hans-Pinsel-Str. 2, 8013 Haar bei München.

**Folgende Preise sind zu gewinnen:**

**1. Preis:**
**Ein PC 10**
**(IBM-kompatibel, 256 KByte RAM, zwei Diskettenlaufwerke mit je 360 KByte Speicherkapazität, MS/DOS-Betriebssystem, Centronics- und V.24-Schnittstelle und Monitor: Wert 5500 Mark)**

**2. bis 11. Preis:**
**je ein Diskettenlaufwerk 1541 für den Commodore 64 oder VC 20**

**12. bis 50. Preis:**
**je ein Buchgutschein im Wert von 50 Mark. Einzulösen nur beim Verlag Markt & Technik AG.**

**Fragen zum Floppy-Laufwerk**

1. Wie groß ist die Speicherkapazität einer formatierten Diskette (in KByte)? ________
2. Wieviel Floppy-Laufwerke sind an den C 64 anschließbar? ________
3. Wieviel mal schneller ist das Floppy-Laufwerk gegenüber der Datasette? ☐ doppelt ☐ 3mal ☐ 10mal
4. Für das Floppy-Laufwerk gibt es mehr als ☐ 500 ☐ 3000 ☐ 5000 Programme.
5. Warum nennt man die Floppy 1541 ein »intelligentes« Laufwerk?
   ☐ wegen eigenem Netzteil ☐ wegen dem DOS
   ☐ weil ein Disk-Controller eingebaut ist
6. Gibt es Lernsoftware für verschiedene Sprachen sowie für Mathematik und Rechtschreibung?
   ☐ ja, schon jetzt ☐ demnächst ☐ nein
7. Welche Dateitypen kennt das Floppy-Laufwerk 1541?
   ☐ TAD ☐ SEQ ☐ REL ☐ UNIX ☐ USR ☐ PRG
   ☐ DEL ☐ ABS ☐ COM

**Allgemeine Fragen**

Welche Computer besitzen Sie? ________

Welche Computer benutzen Sie? ________

Welche Computer wollen Sie sich (noch) kaufen? ________

________

Warum wollen Sie diese(n) Computer kaufen? ________

________

Welche Peripherie besitzen Sie (wollen Sie kaufen)?
☐ Besitze ich:
○ Will ich kaufen:

☐ ○ Drucker Typ? ________
☐ ○ Interface Typ? ________
☐ ○ Plotter Typ? ________
☐ ○ Schreibmaschine/Typenraddrucker Typ? ________
☐ ○ Monitor ☐ S/W ☐ Farbe Typ? ________
☐ ○ Fernseher ☐ S/W ☐ Farbe Typ? ________
☐ ○ Kass.-Recorder Typ? ________
☐ ○ Floppy-Laufwerk Typ? ________
☐ ○ Akustikkoppler/Modem Typ? ________
☐ ○ Joystick Typ? ________
☐ ○ Sonstige Typ? ________

________

**Welche Software (professionell) besitzen/benutzen/interessiert Sie?**

| | Besitze ich | Benutze ich | Interessiert mich |
|---|---|---|---|
| **Programmiersprachen:** | | | |
| ☐ Basic | ☐ | ☐ | ☐ |
| ☐ Comal | ☐ | ☐ | ☐ |
| ☐ Pascal | ☐ | ☐ | ☐ |
| ☐ Assembler | ☐ | ☐ | ☐ |
| ☐ ________ | ☐ | ☐ | ☐ |
| ☐ ________ | ☐ | ☐ | ☐ |
| **Basic-Erweiterungen** | | | |
| ☐ Simons Basic | ☐ | ☐ | ☐ |
| ☐ ________ | ☐ | ☐ | ☐ |
| ☐ ________ | ☐ | ☐ | ☐ |
| **Spiele** | | | |
| ☐ Action | ☐ | ☐ | ☐ |
| ☐ Abenteuer | ☐ | ☐ | ☐ |
| ☐ Sport/Geschicklichk. | ☐ | ☐ | ☐ |
| ☐ Denkspiele | ☐ | ☐ | ☐ |
| ☐ Sonstige ________ | ☐ | ☐ | ☐ |
| **Textverarbeitung** | | | |
| ☐ ________ | ☐ | ☐ | ☐ |
| ☐ ________ | ☐ | ☐ | ☐ |



# Umfragewettbewerb

Grafik

☐ ______ ☐ ☐ ☐

☐ ______ ☐ ☐ ☐

Dateiverwaltung

☐ ______ ☐ ☐ ☐

☐ ______ ☐ ☐ ☐

Sonstige

☐ ______ ☐ ☐ ☐

☐ ______ ☐ ☐ ☐

Ich programmiere selbst
☐ intensiv ☐ gelegentlich ☐ (fast) nie

hauptsächlich in
☐ Basic ☐ Assembler ☐ ______

Welche Themen interessieren Sie besonders ☐, sind Ihrer Meinung nach in der 64'er unterrepräsentiert ○?

☐ ○ Schule/Ausbildung
☐ ○ Messen, Steuern, Regeln
☐ ○ kaufmänn. Anwendung
☐ ○ techn./wiss. Anwendung
☐ ○ private Anwendung
☐ ○ Programmieren lernen
☐ ○ Programmiersprachen
☐ ○ Aktuelle Meldungen
☐ ○ Marktübersichten
☐ ○ Themen für Anfänger
☐ ○ Lernsoftware
☐ ○ Selbstbau
☐ ○ Grafik
☐ ○ Tips & Tricks
☐ ○ Software-Tests
☐ ○ ______
☐ ○ ______

☐ ○ Hardware-Tests
☐ ○ Spiele-Tests
☐ ○ Neue Produkte
☐ ○ Problemlösungen
☐ ○ Themen für Profis
☐ ○ Spiele
☐ ○ DFÜ
☐ ○ Listings
☐ ○ Kurse
☐ ○ Humor
☐ ○ Knobelecke
☐ ○ Interviews
☐ ○ Wettbewerbe
☐ ○ Leserforum
☐ ○ ______
☐ ○ ______

Wie oft kaufen Sie die 64'er?
☐ bin Abonnent ☐ jeden Monat ______ mal pro Jahr
☐ leihe sie mir

Wo kaufen Sie die 64'er?
☐ an irgendeinem Kiosk ☐ immer am gleichen Kiosk
☐ Bahnhofsbuchhandel ☐ Kaufhaus ☐ Computerfachgeschäft

Wieviele Personen lesen — außer Ihnen — noch die 64'er?
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 und mehr

Wieviel Geld geben Sie pro Monat für Literatur zum Computer aus (in Mark)?

______ für Zeitschriften

______ für Bücher

Welche Zeitschriften kaufen Sie noch außer der 64'er?

______

Welche Zeitschriften lesen Sie noch außer der 64'er?

______

Könnten Sie auf die 64'er verzichten?
☐ auf keinen Fall ☐ möglicherweise ☐ ja

Wieviel Geld würden Sie maximal für eine gute Computerfachzeitschrift ausgeben?
☐ 5 Mark ☐ bis 7 Mark ☐ bis 8 Mark ☐ über 8 Mark

Wieviel Geld geben Sie für Ihr Computerhobby aus (in Mark)?
______ ☐ pro Monat ☐ pro Jahr

Kaufen Sie auch Sonderhefte?
☐ nur 64'er-Sonderhefte ☐ auch andere

Wieviele Sonderhefte pro Jahr würden Sie kaufen?
☐ 3 ☐ 5 ☐ 7 ☐ 9 ☐ 11 und mehr

Welche Themen interessieren Sie bei Sonderheften?

______

**Persönliche Daten**

Alter ______ ☐ männlich ☐ weiblich

Stellung im Beruf ______

Schulbildung ______

Eigene Einschätzung
☐ Anfänger ☐ Fortgeschrittener ☐ Profi

Die Adresse wird getrennt von den statistischen Erhebungen gehalten und dient nur zur Ermittlung der Gewinner.

Anschrift:

Name: ______

Vorname: ______

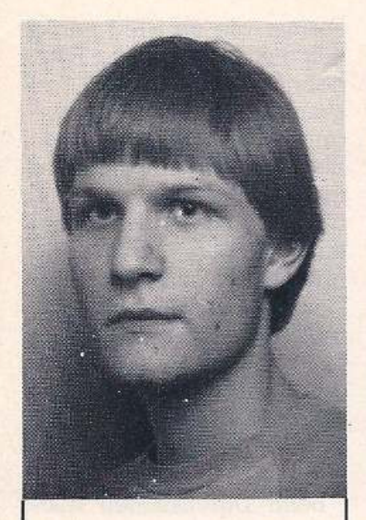
Straße: ______

Ort: ______

Einsendeschluß ist der 1. Juli 1985.

Der Rechtsweg ist ausgeschlossen.





# »Dokumentationshilfe«: Cross-Referenz-Liste C 64

**Die Dokumentation von Basic-Programmen ist manchmal eine mühevolle Aufgabe. Trotzdem ist sie sinnvoll oder sogar notwendig. Aus diesem Grund schlugen wir einen Programmierwettbewerb zur Dokumentationshilfe vor, dessen Ergebnis hier vor Ihnen liegt.**

## Lebenslauf

Angefangen hat meine Computerbegeisterung mit etwa zehn Jahren. Damals schenkte meine Großmutter mir einige Spiele von Atari. Diese Spiele waren aber nur kurzzeitig von Interesse.

Ich kaufte mir fast alle Computer-Zeitschriften und konnte in den Kaufhäusern nicht an den Computern vorbeigehen, die dort ausgestellt waren.

Als mein Vater sich für seine Handelsvertretung einen Commodore-Computer zulegte, hatte ich Gelegenheit, durch meine Vorkenntnisse, die Gespräche zwischen einer Leasingfirma und meinem Vater zu beeinflussen.

Meine Großmutter unterstützte mein Computerhobby auch finanziell und ich war mit 15 Jahren in der Lage, mir meinen eigenen C 64 mit Floppy zu kaufen. Nach den großen Ferien komme ich in die gymnasiale Oberstufe. Als Abiturfächer werde ich Physik, Mathematik und Englisch wählen, da ich Informatik studieren möchte. Diesen Beruf würde ich gern in der Forschung ausüben.

Zur Zeit überarbeite ich das EDV-Programm der Firma meines Vaters, um es bedienungsfreundlicher zu gestalten und dieses auch für einen Tennisclub von etwa 500 Mitgliedern (Verwaltung, Mitteilungen, Beitragseinzug) zu verwenden.

(Stefan Becker)

Jeder Basic-Programmierer kennt das Problem: Man schreibt lange und auch komplizierte Programme und merkt, daß langsam aber sicher der Überblick verloren geht. Oder auch, wenn versucht wird, ein fremdes, nicht selbst erstelltes Programm zu verstehen: Was bedeutet diese Variable, und wo taucht sie im Programm noch auf? Von wo und wie oft wird diese Programmzeile angesprungen?

Bei gut strukturierten Programmen findet man relativ schnell durch, aber die sind selten zu finden. Doch selbst dann ist eine gute Beschreibung eine hilfreiche Sache und erleichtert den späteren Wiedereinstieg.

Eine Crossreferenzliste unterstützt bei der Erstellung einer Dokumentation. Gerade bei kommerziellen Software-Projekten wird nicht auf sie verzichtet. Sie enthält dabei nicht nur eine Aufzählung aller Sprünge (sofern bei höheren Programmiersprachen überhaupt vorhanden), sondern vor allem eine komplette Variablenliste. Und diese Funktionen hat auch unser Programm. Im einzelnen enthält sie:

1. Liste aller Zeilen, in denen Sprungbefehle enthalten sind. Angegeben wird die Zeilennummer und die angesprungene Zeile.
2. Liste aller Zeilen, die angesprungen werden. Angegeben sind die Zeilennummer und alle Zeilen, von denen aus diese Zeile angesprungen wird. Eine sehr nützliche Information.
3. Eine Liste aller im Programm vorkommenden Variablen. Angegeben wird, alphabetisch sortiert, der Variablenname und die Zeilen, in denen sie vorkommt. Zusätzlich hat man die Möglichkeit, zu jeder Variablen einen kurzen Kommentar mit einzugeben.
4. Liste aller Variablen mit dem vorhin genannten Kommentar, aber ohne Hinweise auf die Zeilen, in denen sie vorkommen.

Mit diesen Informationen kann man schon etwas anfangen. Doch jetzt etwas für Programmierer selbst:

## Wie funktioniert das?

Wie sucht man nach Variablen? Wie nach Sprungbefehlen? Wie so oft, ist die Lösung ganz einfach.

Wie ein Basic-Programm im Speicher steht, wurde ausführlich im 64'er, Ausgabe 2/85, Seite 87, beschrieben. Fast in der gleichen Form wird ein Programm auch auf Diskette gespeichert. Lediglich zwei Byte für die Ladeadresse des Programms kommen dazu. Beim absoluten Laden eines Programms (LOAD"NAME",8,1) holt sich der Basic-Interpreter diese zwei Byte, um festzustellen, an welcher Stelle im Speicher das Programm beginnt. Beim normalen LOAD-Befehl, also ohne die ",1" am Ende, wird jedes Programm ab Basic-Anfang gesetzt, also ab $801 (= dezimal 2049).

An einem kleinen Beispiel soll gezeigt werden, wie man sich alle Zeilen anzeigen lassen kann, die den Befehl GOTO enthalten. Sehen Sie sich dazu das Flußdiagramm (Bild 1) an.

Als erstes muß eine »Programmdatei« geöffnet werden. Das geht zum Beispiel mit OPEN2,8,2,"NAME,P,R". »P« steht dabei für Programm und »R« für READ, also lesen.

Dann folgt das Lesen der Byte. Dazu nimmt man den GET#-Befehl. Der INPUT #-Befehl ist zwar wesentlich schneller, »schafft« aber nur maximal 88 Zeichen, benötigt ein Carriage Return (CHR$(13)) als Endekennzeichen (bei Programmen sind es Null-Byte) und verträgt keine Kommata (in Basic-Programmen oft vorhanden). Außerdem erlaubt uns der GET#-Befehl die Überprüfung einzelner Zeichen direkt nach dem Laden.

Lesen wir jetzt also die ersten Byte eines Basic-Programms. Am Anfang, das sind die ersten zwei Bytes, steht die Startadresse. Bei Basic-Programmen ist sie 2049, hexadezimal ausgedrückt $0801. Darauf folgen zwei Byte für die Anfangsadresse der nächsten Basic-Zeile (gemeint ist hier die Adresse im RAM, die natürlich auch gespeichert wird). Erst jetzt beginnt die wirkliche Basic-Zeile: Zwei Byte für die Zeilennummer, aufgetrennt in Low- und High-Byte, gefolgt vom Basic-Text. Das Ende einer Basic-Zeile erkennen Sie (und der Basic-Interpreter) an einem Null-Byte. Danach folgt wieder die Anfangsadresse der nächsten Basic-Zeile (Low- und High-Byte) und der nächsten Zeilennummer (auch Low- und High-Byte), danach der Basic-Text, etc., etc.. Das Ende eines Basic-Programms ist gekennzeichnet durch drei aufeinanderfolgende Null-Byte, und auch

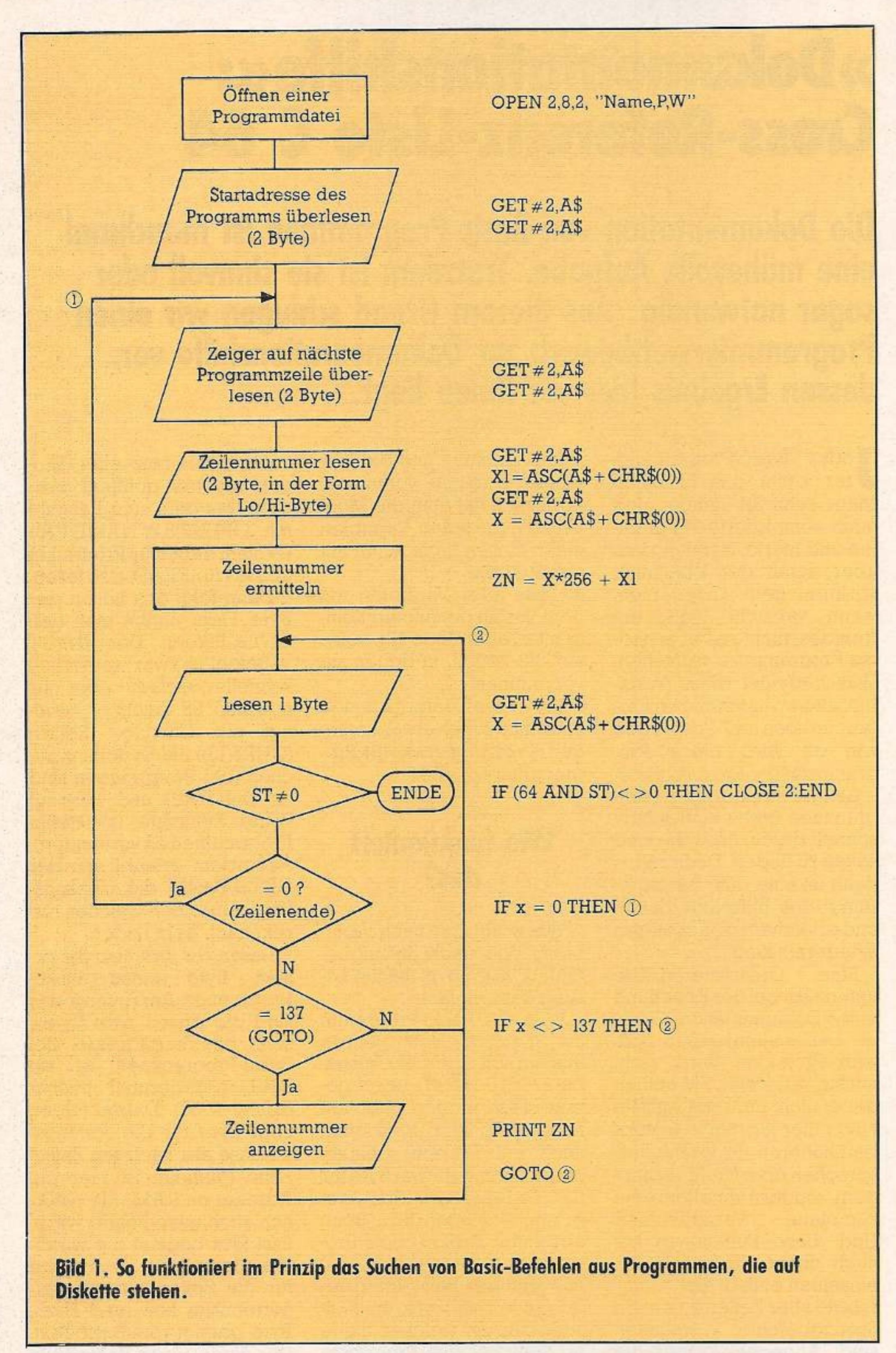



**Bild 1. So funktioniert im Prinzip das Suchen von Basic-Befehlen aus Programmen, die auf Diskette stehen.**

die Statusvariable ST wird auf 64 gesetzt.

Das war's schon. Mit diesem Wissen sind Sie in der Lage, ein Basic-Programm komplett zu analysieren. Eine vollständige Liste aller Token, das sind die Abkürzungen, die Codes der Basic-Befehle, finden Sie in Bild 2.

Doch nun zum Programm »Cross-Ref 64«.

## Cross-Ref 64

Das hier vorgestellte Programm Cross-Ref 64 (Listing 2) arbeitet auf allen Commodore-Computern, mit einem Floppy- oder Kassetten-Laufwerk und einem Drucker.

Nach dem Start mit RUN kann die Ausgabe der fertigen Listen auf Drucker oder Bildschirm gewählt werden (siehe auch Zeile 200 bis 260). Die Variable DV enthält die jeweils gültige Gerätenummer. Danach müssen Sie den Dateinamen des zu durchsuchenden Files eingeben.

Ihr Computer liest sich nun das Programm wie ein sequentielles File Byte für Byte durch. Das Programm steht auf der Diskette im gleichen Format wie im RAM, lediglich die ersten beiden Bytes geben den Start des Programms an. Liegt Ihr Programm nicht ab 2049, zum Beispiel bei Maschinenprogrammen üblich, so bricht das Programm die Bearbeitung ab. Sollten Sie das Programm auf einem anderen Commodore-Computer betreiben, so geben Sie einfach Listing 3 ein und tauschen die auf dem Bildschrim erscheinende Zahl gegen die in Zeile 290 stehende 2049 aus.

Beim Durchsuchen »hangelt« sich der Computer durch die Zeilen, indem er, immer wenn er auf einen Charakter 0 trifft, eine neue Zeile beginnt. Im ersten Paß durchsucht er das Programm auf die Token für die Sprungbefehle GOTO, RUN, GOSUB und THEN (Zeile 330). Sie haben, in der gleichen Reihenfolge, die ASCII-Werte 137, 138, 141 und 167. Gefundene Sprünge werden auf zwei Arten gespeichert. Als erstes im Format ZEILENNUMMER : SPRUNGZIEL (Bild 3) in den Zeilen 400 bis 420, darauf im Format SPRUNGZIEL : ZEILENNUMMERN (Bild 3). Wobei SPRUNGZIEL immer die Zahl ist, die hinter GOTO oder THEN steht. ZEILENNUMMER ist jeweils die Zeile, in der der Sprungbefehl auftaucht.

Die Zeilennummern werden in den Feldern ps$( und sp$( gespeichert. Wird in einem der Felder die Maximallänge von 70 Zeichen überschritten, so wird ein neues Feld angelegt (Zeile 410 und 450).

Haben Sie den ersten Pass glücklich überstanden, so wird die Liste auf das von Ihnen gewählte Gerät übertragen (Zeile 580 bis 680). Die aufsteigend sortierten Zeilennummern werden so ausgegeben, daß nie zwei gleiche Zeilennummern untereinanderstehen. Statt dessen werden sechs Leerzeichen gedruckt (Zeile 660).

Der zweite Pass läuft im Prinzip wie der erste ab. Die Variablen mit zugehörigen Zeilennummern werden lediglich im Feld VA$( gespeichert und sortiert. Selbstverständlich wird der Text hinter REM und DATA sowie An-

führungszeichen überlesen (Zeile 750-790).

In diesem Stadium kommt Arbeit auf Sie zu. Jetzt haben Sie die Möglichkeit, die Variablen mit einem Text zu versehen. Dabei müssen Sie sich allerdings auf 25 Zeichen beschränken.

Antworten Sie auf die Frage »WOLLEN SIE ZU DEN VARIABLEN BEMERKUNGEN EINGEBEN« mit »J«, so können sie mit Cursor UP und DOWN durch die Liste wandern. Die momentane Variable wird jeweils angezeigt. Haben Sie die richtige Stelle gefunden, so drücken Sie »RETURN« und können Ihren Text eingeben. Durch nochmaliges Drücken der »RETURN«-Taste wird der Text gespeichert.

Haben Sie Ihre Texte verteilt, so beginnt das Programm nach Drücken der »N«-Taste zu drucken (Zeilen 1180 bis 1280). Auch hier können Sie den Druckvorgang wiederholen (Bild 4 und 5).

**Umstellungshinweise**

Dieses Programm ist auf allen Computern lauffähig, die ihren Basic-Text nach Commodoreart speichern. Potentielle »Umschreiber« können sich daher auf die Zeile 330 beschränken. Auch die Druckausgabe ist leicht modifizierbar. Die Sekundäradresse der OPEN-Befehle in den Zeilen 580 und 1180 stellen den Epson MX-80 über das Print-64-Interface auf Klein/Großschrift um. Dies kann auch durch normale »OPENs« mit nachfolgenden Steuercodes erfolgen.

Sogar bei einem Kassettenlaufwerk ist das Programm verwendbar. Auch hier ändern Sie in den Zeilen 290 und 730 die OPEN-Befehle (Listing 4).

(Stefan Becker/gk)

| Befehl | Token DEZ | Token HEX | Befehl | Token DEZ | Token HEX | Befehl | Token DEZ | Token HEX |
|---|---|---|---|---|---|---|---|---|
| END | 128 | 80 | CONT | 154 | 9A | SGN | 180 | B4 |
| FOR | 129 | 81 | LIST | 155 | 9B | INT | 181 | B5 |
| NEXT | 130 | 82 | CLR | 156 | 9C | ABS | 182 | B6 |
| DATA | 131 | 83 | CMD | 157 | 9D | USR | 183 | B7 |
| INPUT# | 132 | 84 | SYS | 158 | 9E | FRE | 184 | B8 |
| INPUT | 133 | 85 | OPEN | 159 | 9F | POS | 185 | B9 |
| DIM | 134 | 86 | CLOSE | 160 | A0 | SQR | 186 | BA |
| READ | 135 | 87 | GET | 161 | A1 | RND | 187 | BB |
| LET | 136 | 88 | NEW | 162 | A2 | LOG | 188 | BC |
| GOTO | 137 | 89 | TAB | 163 | A3 | EXP | 189 | BD |
| RUN | 138 | 8A | TO | 164 | A4 | COS | 190 | BE |
| IF | 139 | 8B | FN | 165 | A5 | SIN | 191 | BF |
| REST. | 140 | 8C | SPC | 166 | A6 | TAN | 192 | C0 |
| GOSUB | 141 | 8D | THEN | 167 | A7 | ATN | 193 | C1 |
| RETURN | 142 | 8E | NOT | 168 | A8 | PEEK | 194 | C2 |
| REM | 143 | 8F | STEP | 169 | A9 | LEN | 195 | C3 |
| STOP | 144 | 90 | + | 170 | AA | STR$ | 196 | C4 |
| ON | 145 | 91 | — | 171 | AB | VAL | 197 | C5 |
| WAIT | 146 | 92 | * | 172 | AC | ASC | 198 | C6 |
| LOAD | 147 | 93 | / | 173 | AD | CHR$ | 199 | C7 |
| SAVE | 148 | 94 | ↑ | 174 | AE | LEFT$ | 200 | C8 |
| VERIFY | 149 | 95 | AND | 175 | AF | RIGHT$ | 201 | C9 |
| DEF | 150 | 96 | OR | 176 | B0 | MID$ | 202 | CA |
| POKE | 151 | 97 | > | 177 | B1 | GO | 203 | CB |
| PRINT# | 152 | 98 | = | 178 | B2 | | | |
| PRINT | 153 | 99 | < | 179 | B3 | | | |

**Bild 2. Basic-Befehlswörter und deren Token.**

```
Programmname: test

sprungtabelle
---------------------------

zeile :    sprung auf zeile

   140:  1000  2000  3000

zeile :    wird angesprungen von

  1000:   140
  2000:   140
  3000:   140
```

**Bild 3. Diese Sprungtabelle wird ausgegeben, wenn das Programm »Listing 1« untersucht wird.**

```
Variablenliste ohne Zeilennummer
---------------------------------
a$(    :          eingegebene Werte
aw     :      Parameter ON-Anweisung
aw$    :                   siehe aw
b(     :               keine Ahnung
c%(    :                    wie b(
i      :
j      :
na$    :                  Dateiname
x      :             Hilfsvariable
```

**Bild 4. Diese Variablenliste erstellt »Cross-Ref 64« nach Untersuchung von »Listing 1«. Die Kommentare müssen natürlich zusätzlich eingegeben werden.**

```
liste der variablen         :
----------------------------
a$(    :    10    40    50   100
aw     :   130   140  1000  2000  3000
aw$    :   130
b(     :    10    20
c%(    :    10
i      :    20   100
j      :    30    40    50    60    90
na$    :    70    80
x      :    90  1000  2000  3000
```

**Bild 5. Auch eine Variablenliste ohne Zeilennummern ist möglich.**

```
5 REM X=2
10 DIMA$(100),B(20),C%(20)
20 FORI=1TO20:B(I)=I:NEXT
30 FORJ=1TO20
40     PRINT" WERT NR."J;:INPUTA$(J)
50     IF A$(J)="X" THEN J=20
60 NEXT J
70 INPUT"PRG.NAME = "; NA$
80 OPEN2,8,2,NA$+"P,W"
90 FORX=1TOJ
100      PRINT#2,A$(I)
110 NEXT
120 CLOSE2
130 INPUT"AUSWAHL =";AW$:AW=VAL(AW$)
140 ON AW GOSUB 1000,2000,3000
150 END
1000 PRINTAW:X=1:RETURN
2000 PRINTAW:X=2:RETURN
3000 PRINTAW:X=3:RETURN
```

**Listing 1. Dieses Programm liefert die Listen Bild 3 bis 5.**

```
10 REM*********************************    <003>
20 REM* PROGRAMMNAME : XREF           *    <236>
30 REM*      C-64                     *    <219>
40 REM*      FLOPPY 1541 O. AEHNLICHE *    <174>
50 REM*      DRUCKER (Z.B. MPS 801)   *    <247>
60 REM* VON STEFAN BECKER             *    <127>
70 REM*********************************    <063>
80 CLR:GOTO 150                            <069>
90 GET#1,A$:X=ASC(A$+N$):IF(64 AND ST)=0 T
   HEN RETURN                              <169>
100 CLOSE 1:IF PA THEN PA=0:GOTO 520       <255>
110 GOTO 960                               <150>
120 GOSUB 90:X1=X:GOSUB 90:X=X1+256*X:RETU
    RN                                     <075>
130 IF SP<OB AND VA<OB AND PS<OB THEN RETU
    RN:REM*** GRENZEN ERREICHT ? ***       <188>
140 PRINT:PRINT"BITTE DIE VARIABLE OB IN Z
    EILE 150 VERGROESSERN.":GOTO 1310      <222>
150 OB=500:DIM SP$(OB),PS$(OB),VA$(OB)     <117>
160 REM*** OB IST OBERGRENZE DER ANZAHL DE
    R SPRUENGE UND VARIABLEN ***           <247>
170 POKE 53280,6:POKE 53281,6              <219>
180 N$=CHR$(0)                             <032>
190 LE$="{50SPACE}"                        <105>
200 PRINT CHR$(147)CHR$(9)CHR$(14)CHR$(8)C
    HR$(144);                              <031>
210 PRINT"{RVSON,10SPACE}CROSS-REFERENZ-LI
    STER{9SPACE}"                          <089>
220 PRINT"{2DOWN}AUSGABE AUF ":PRINT"{DOWN
    ,RVSON}B{RVOFF}ILDSCHIRM ODER{SPACE,RV
    SON}D{RVOFF}RUCKER{3UP}"               <109>
230 PRINT SPC(12);:POKE 204,0              <175>
240 GET A$:IF A$<>"B"AND A$<>"D"GOTO 240   <041>
250 IF A$="B"THEN PRINT"BILDSCHIRM.":DV=3:
    GOTO 270                               <074>
260 PRINT"DRUCKER.":DV=4                   <090>
270 PRINT"{DOWN,23SPACE,UP}"               <142>
280 OPEN 1,0:PRINT"PROGRAMMNAME: ";:INPUT#
    1,NA$:CLOSE 1:PRINT                    <131>
290 OPEN 2,8,15:OPEN 1,8,2,NA$+",P,R":GOSU
    B 1290:GOSUB 120:IF X=2049 THEN 310    <067>
300 PRINT:PRINT"DAS PROGRAMM MUSS AB 2049
    LIEGEN.":GOTO 1310                     <030>
302 :                                      <105>
304 :                                      <107>
305 REM*********************************   <001>
306 REM      PASS 1 SPRUNGTABELLE          <002>
308 REM*********************************   <004>
309 :                                      <112>
310 PRINT"{CLR}PASS 1 (SUCHEN DER SPRUENGE
    ){DOWN}":PA=1                          <169>
320 GOSUB 120:GOSUB 120:ZE$=RIGHT$("{5SPAC
    E}"+STR$(X),5):PRINT"{HOME,2DOWN}"ZE$  <079>
330 GOSUB 90:IF X=137 OR X=138 OR X=141 OR
     X=167 THEN S1$="":GOTO 360            <074>
340 IF X=0 GOTO 320                        <056>
350 GOTO 330                               <126>
360 GOSUB 90:IF X=32 GOTO 360              <181>
370 IF X>=48 AND X<=57 THEN S1$=S1$+A$:GOT
    O 360                                  <017>
380 IF S1$=""GOTO 470                      <202>
390 S1$=RIGHT$("{6SPACE}"+S1$,6)           <243>
400 IF LEFT$(SP$(SP),5)<>ZE$THEN SP=SP+1:G
    OSUB 130:SP$(SP)=ZE$+":"               <223>
410 IF LEN(SP$(SP))>70 THEN SP=SP+1:GOSUB
    130:SP$(SP)=ZE$+":"                    <116>
420 IF RIGHT$(SP$(SP),6)<>S1$THEN SP$(SP)=
    SP$(SP)+S1$                            <148>
430 FOR I=1 TO PS                          <163>
440 IF LEFT$(PS$(I),6)<>S1$THEN NEXT:PS=I:
    GOSUB 130:PS$(I)=S1$+":"               <165>
450 IF LEN(PS$(I))>70 THEN NEXT:PS=PS+1:I=
    PS:GOSUB 130:PS$(I)=S1$+":"            <097>
460 IF RIGHT$(PS$(I),5)<>ZE$THEN PS$(I)=PS
    $(I)+" "+ZE$                           <209>
470 IF X=0 GOTO 320                        <186>
480 IF X=44 THEN S1$="":GOTO 360           <127>
490 IF X=58 OR X>=127 OR(X>=65 AND X<=90)G
    OTO 330                                <013>
500 PRINT:PRINT"{DOWN}FEHLER IM QUELLPROGR
    AMM. ZEILE:"ZE$:GOTO 1310              <150>
502 :                                      <049>
504 :                                      <051>
506 REM ------------------------------     <253>
510 REM*** SORTIEREN DER SPRUNGZIELE (FELD
     PS$( ***                              <170>
515 REM ------------------------------     <007>
516 :                                      <064>
520 FOR I=1 TO PS                          <254>
530 FOR J=I TO PS                          <033>
540 IF LEFT$(PS$(I),5)<LEFT$(PS$(J),5)GOTO
     560                                   <055>
550 PS$(0)=PS$(I):PS$(I)=PS$(J):PS$(J)=PS$
    (0)                                    <200>
560 NEXT J                                 <254>
570 NEXT I                                 <007>
574 :                                      <122>
575 REM-----------------------------       <112>
576 REM   AUSGABE SPRUENGE + SPRUNGZIELE   <181>
577 REM-----------------------------       <114>
578 :                                      <126>
580 OPEN 1,DV,7:PRINT#1,"PROGRAMMNAME: "NA
    $:PRINT#1                              <168>
590 PRINT#1,"SPRUNGTABELLE{11SPACE}"       <097>
600 PRINT#1,"------------------------"     <037>
605 PRINT#1,"ZEILE :{3SPACE}SPRUNG AUF ZEI
    LE{6SPACE}":PRINT#1                    <130>
610 FOR I=1 TO SP                          <088>
620 IF LEFT$(SP$(I-1),5)=LEFT$(SP$(I),5)TH
    EN PRINT#1,SPC(7)MID$(SP$(I),7):GOTO 6
    40                                     <027>
630 PRINT#1," "SP$(I)                      <018>
640 NEXT I:PRINT#1                         <080>
645 PRINT#1,"ZEILE :{3SPACE}WIRD ANGESPRUN
    GEN VON":PRINT#1                       <044>
650 PS$(0)="":FOR I=2 TO PS                <249>
660 IF LEFT$(PS$(I-1),6)<>LEFT$(PS$(I),6)T
    HEN PRINT#1,PS$(I):GOTO 680            <119>
670 PRINT#1,SPC(6);MID$(PS$(I),7)          <180>
680 NEXT I:CLOSE 1                         <128>
690 PRINT"{DOWN,RVSON}N{RVOFF}OCHMALS/{RVS
    ON}W{RVOFF}EITER"                      <062>
700 GET A$:IF A$<>"N"AND A$<>"W"GOTO 700   <023>
710 IF A$="N"GOTO 580                      <034>
712 :                                      <004>
714 :                                      <006>
715 REM*******************************     <156>
716 REM      PASS 2 VARIABLE SUCHEN        <210>
718 REM*******************************     <159>
719 :                                      <011>
720 PRINT"{CLR}PASS 2 (SUCHEN DER VARIABLE
    N){DOWN}"                              <194>
730 OPEN 1,8,2,NA$+",P,R":GOSUB 1290:GOSUB
     120                                   <034>
740 GOSUB 120:GOSUB 120:ZE$=RIGHT$("{5SPAC
    E}"+STR$(X),5):PRINT"{HOME,2DOWN}"ZE$:
    Y=0                                    <105>
750 GOSUB 90                               <230>
760 IF X=0 GOTO 740                        <227>
770 IF X=34 OR X=131 OR X=143 GOTO 800     <195>
780 IF X>64 AND X<91 GOTO 850              <087>
790 GOTO 750                               <062>
800 IF X=131 THEN Y=1                      <016>
810 GOSUB 90:IF X=0 GOTO 740               <070>
820 IF X=34 GOTO 750                       <088>
830 IF X=58 AND Y=1 GOTO 750               <083>
840 GOTO 810                               <109>
850 V1$=A$                                 <023>
860 GOSUB 90                               <085>
870 IF X=36 OR X=37 OR(X>47 AND X<58)OR(X>
    64 AND X<91)THEN V1$=V1$+A$:GOTO 860   <222>
880 IF X=40 THEN V1$=V1$+A$                <042>
890 V1$=LEFT$(V1$+"{4SPACE}",4)            <236>
900 FOR I=1 TO VA                          <111>
910 IF LEFT$(VA$(I),4)<>V1$THEN NEXT:VA=I:
    GOSUB 130:VA$(I)=V1$+"{2SPACE}:"       <093>
920 IF LEN(VA$(I))>50 THEN NEXT:VA=VA+1:I=
    VA:GOSUB 130:VA$(I)=VA$+"{2SPACE}:"    <014>
930 IF RIGHT$(VA$(I),5)<>ZE$THEN VA$(I)=VA
    $(I)+" "+ZE$                           <133>
940 IF X=0 GOTO 740                        <152>
950 GOTO 750                               <222>
952 :                                      <245>
954 :                                      <247>
956 REM ----------------------------       <193>
960 REM *** SORTIEREN DER VARIABLEN (FELD
    VA$( ***                               <158>
965 REM ----------------------------       <202>
968 :                                      <005>
970 FOR I=1 TO VA                          <181>
980 FOR J=I TO VA                          <216>
990 IF LEFT$(VA$(I),4)>LEFT$(VA$(J),4)GOTO
     1010                                  <005>
1000 VA$(0)=VA$(I):VA$(I)=VA$(J):VA$(J)=VA
     $(0)                                  <067>
1010 NEXT J                                <193>
1020 NEXT I:VA=VA-1:I=VA                   <140>
1025 :                                     <063>
```

**Listing 2. Cross-Referenz-Liste 64**

```
1030 PRINT"{CLR}WOLLEN SIE ZU DEN VARIABLE
     N":PRINT"BEMERKUNGEN EINGEBEN?{2SPACE
     }";                                  <071>
1040 GET A$:IF A$="N"GOTO 1180            <219>
1050 IF A$<>"J"GOTO 1040                  <078>
1060 PRINT"JA{3DOWN}"                     <067>
1070 PRINT"{HOME,6DOWN}VARIABLENNAME: ";LE
     FT$(VA$(I),4)                        <000>
1073 PRINT LE$                            <131>
1075 IF LEN(VA$(I))>70 THEN PRINT"{2UP}"RI
     GHT$(VA$(I),25)                      <235>
1080 GET A$:IF A$<>"{UP}"AND A$<>"{DOWN}"A
     ND A$<>CHR$(13)GOTO 1080             <184>
1090 IF A$="{UP}"THEN I=I+1:IF I>VA THEN I
     =VA                                  <018>
1100 IF A$="{DOWN}"THEN I=I-1:IF I<1 THEN
     I=1                                  <211>
1110 IF A$<>CHR$(13)GOTO 1070             <123>
1120 VR$=LEFT$(VA$(I),4):OPEN 1,0         <022>
1130 PRINT"TEXT: ";:INPUT#1,TE$:PRINT:CLOSE
      1:TE$=LEFT$(TE$,25)                 <077>
1140 FOR I=VA TO 1 STEP-1                 <229>
1150 IF LEFT$(VA$(I),4)<>VR$THEN NEXT:PRIN
     T"NICHT VORHANDEN.":GOTO 1070        <159>
1160 VA$(I)=LEFT$(VA$(I)+LE$,55)+RIGHT$(LE
     $+TE$,25):GOTO 1030                  <240>
1170 VR$=LEFT$(VR$,4)                     <089>
1174 ;                                    <213>
1175 REM--------------------------------  <157>
1176 REM      AUSGABE VARIABLE            <105>
1177 REM--------------------------------  <159>
1180 PRINT:OPEN 1,DV,7                    <108>
1190 PRINT#1,"{2DOWN}LISTE DER VARIABLEN{5
     SPACE}:"                             <047>
1200 PRINT#1,"-------------------------"  <082>
1210 FOR I=VA TO 1 STEP-1                 <043>
1220 IF LEFT$(VA$(I+1),4)<>LEFT$(VA$(I),4)
     THEN PRINT#1,VA$(I):GOTO 1240        <169>
1230 PRINT#1,SPC(6);MID$(VA$(I),7)        <218>
1240 NEXT I                               <167>
1250 PRINT"{DOWN,RVSON}N{RVOFF}OCHMALS/{RV
     SON}W{RVOFF}EITER"                   <112>
1260 GET A$:IF A$<>"N"AND A$<>"W"GOTO 1260 <123>
1263 IF A$="N"GOTO 1190                   <123>
1266 PRINT"VARIABLENLISTE OHNE ZEILENNUMME
     R (J/N)"                             <197>
1267 GET R$:IF R$=""THEN 1267             <182>
1268 IF R$<>"J"THEN PRINT:PRINT"ENDE":GOTO
      1310                                <230>
1269 PRINT#1:PRINT#1:PRINT#1,"VARIABLENLIS
     TE OHNE ZEILENNUMMER"                <018>
1270 PRINT#1,"----------------------------
     ---"                                 <166>
1271 FOR I=VA TO 1 STEP-1                 <104>
1272 IF LEN(VA$(I))>70 THEN PRINT#1,LEFT$(
     VA$(I),8);RIGHT$(VA$(I),25):GOTO 1274 <068>
1273 PRINT#1,LEFT$(VA$(I),8)              <196>
1274 NEXT I                               <201>
1280 GOTO 1310                            <083>
1290 INPUT#2,A$,B$:IF A$="00"THEN RETURN  <157>
1300 PRINT:PRINT"DISK-ERROR: ";B$         <134>
1310 CLOSE 1:CLOSE 2:END                  <186>
```

**Listing 2. »Cross-Referenz-Liste 64« (Schluß)**

```
10 INPUT"NAME";A$                         <148>
20 OPEN 1,8,2,A$+",P,R"                   <031>
30 GET#1,A$,B$                            <054>
40 CLOSE 1                                <249>
50 PRINT"ANFANGSADRESSE: ";ASC(A$+CHR$(0))
   +256*ASC(B$+CHR$(0))                   <089>
60 :                                      <118>
70 REM FUER KASSETTENBETRIEB:             <240>
80 REM ZEILE 20: OPEN1,1,0,A$             <161>
```

**Listing 3. Cross-Ref 64 für andere Commodore-Computer**

```
290 OPEN 1,1,0,NA$:GOSUB 120:IF X=2049 THE
    N 310                                 <190>
730 OPEN 1,1,0,NA$:GOSUB 120              <158>
999 REM DIE ZEILEN 1290 BIS 1310 ENTFALLEN <026>
```

**Listing 4. Anpassung von Cross-Ref 64 an die Datasette**



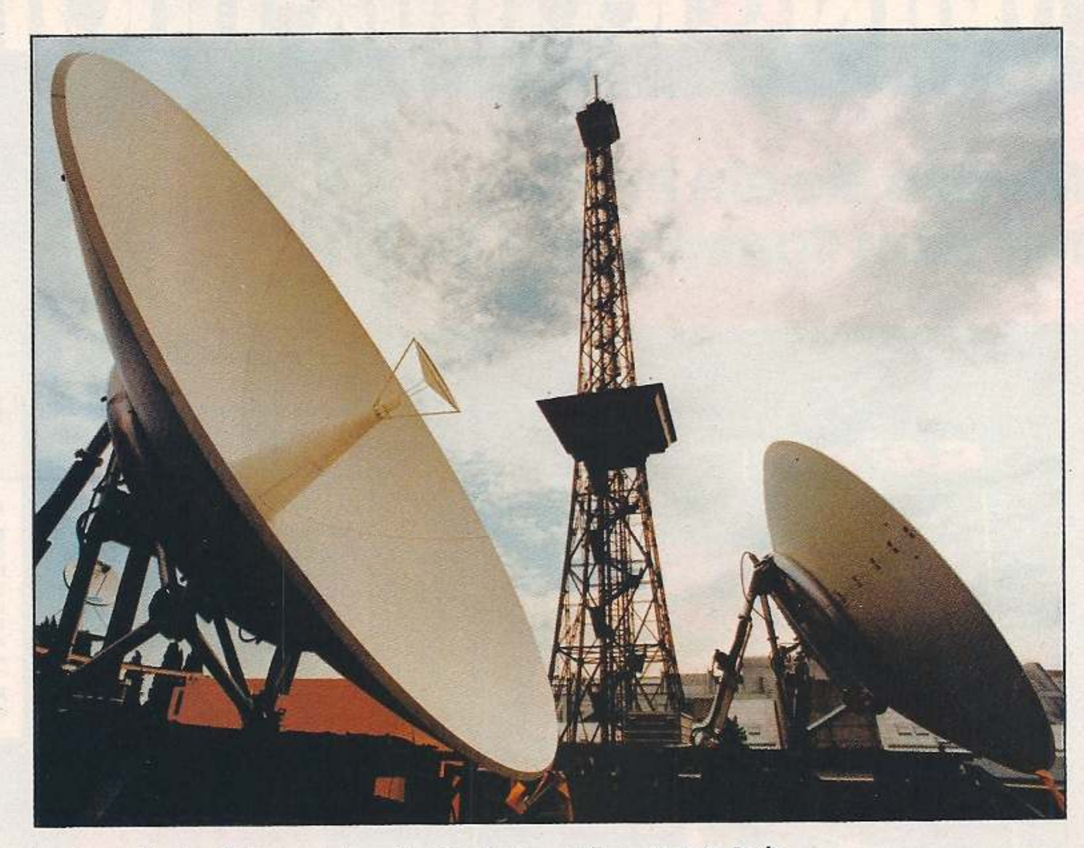
# Gewinnen Sie zwei Btx-Fernsehgeräte Loewe MCP 114 und eine Reise zur Internationalen Funkausstellung '85 in Berlin!

**Gesucht: Vier originelle Spiel- und Lernprogramme für den C 64. Schreiben Sie die erste C 64-Telesoftware für das deutsche Bildschirmtext-System! Jeder der mitmacht hat eine Chance.**



Die Deutsche Bundespost wird bei der Internationalen Funkausstellung 1985 in Berlin erstmals das Fernladen von Computerprogrammen für den C 64 mittels Bildschirmtext demonstrieren. Die vier Programme, die auf diesem Weg von der Btx-Zentraldatenbank in Ulm in fünf Demonstrationsmodelle C 64 in Halle 13 des Internationalen Ausstellungsgeländes am Funkturm geschickt werden, sollen von den 64'er-Lesern geschrieben werden. Gesucht wird nach je zwei originellen Spiel- und zwei Lernprogrammen.

- Der erste Preis ist ein Btx-Fernsehgerät Loewe MCP 114 und eine Reise zur Preisverleihung während der Funkausstellung im September für eine Person.
- Der zweite Preis ist ein Btx-Fernsehgerät Loewe MCP 114.
- Der dritte und vierte Preis ist je ein Drucker MPS 802.



**Gewinnen Sie eine Reise zur Internationalen Funkausstellung 1985 in Berlin**

## Das ist Bildschirmtext

Bildschirmtext ist ein neuer Informationsdienst der Post. Dabei können Informationen aus Btx-Zentralen (Computern) der Post über die Telefonleitung auf dem Bildschirm im Wohnzimmer abgerufen und wegen der Dialogfähigkeit des Systems auch gesendet werden.

Für die Nutzung von Bildschirmtext braucht man ein Farbfernsehgerät mit einer Zusatzeinrichtung (Decoder). Dazu eine Btx-Anschlußbox, die von der Post zwischen Telefon und Fernsehgerät installiert wird. Dort werden die digitalen Signale moduliert und so umgewandelt, daß sie über die Fernsprechleitung der Post-Zentrale zugeführt werden können.

Die Möglichkeiten bei der Nutzung von Bildschirmtext beschränken sich allerdings nicht nur auf den Abruf visueller Informationen (Text, Zeichengrafik, Kleingrafik), sondern man kann auch Daten übermitteln, die einen Bildplattenspieler steuern, die ein Programm für einen Mikrocomputer darstellen oder die Eingangsdaten für ein Computerprogramm.

Die Faszination dieserart Programme in den eigenen Heimcomputer zu laden und dort ablaufen zu lassen liegt nicht zuletzt in der Nutzung eines modernen Telekommunikationsnetzes. Zum anderen kommt man so problemlos an ein umfassendes Programmangebot. Um in der Btx-Zentrale gespeicherte Programme und Daten in den Computer laden zu können (downloading), bedarf es auf der Benutzerseite neben dem Rechner, wie beim normalen Empfang von Btx-Informationsseiten eines Btx-Fernsehgerätes (mit Btx-Anschluß), einer Verbindung zwischen Rechner und Fernsehgerät und der Betriebssoftware für den Rechner zur Steuerung der Kommunikation und des Ladevorganges.

**Folgende Bedingungen müssen die Programme erfüllen:**

* Das Programm kann in Basic oder Maschinensprache geschrieben sein
* Die maximale Länge beträgt 7 KByte, oder 30 Blocks auf der Diskette
* Das Programm muß vollständig im Speicher stehen, das heißt es dürfen keine weiteren Programmteile nachgeladen werden
* Der Speicherbereich zwischen $C000 und $CFFF und die Puffer in der Zeropage dürfen nicht benutzt werden
* Das Programm darf nicht über einen Autostart verfügen, das heißt, es darf sich nach dem Laden nicht selbst starten
* Es sollte eine Bedienungsanleitung im Programm wahlweise angesprungen werden können
* Spielprogramme **müssen** über Joysticks gesteuert werden
* Während des Programmlaufs dürfen keine Zugriffe auf Diskette oder Kassette erfolgen, das heißt es dürfen weder Daten nachgeladen noch abgespeichert werden
* Die schriftliche Bedienungsanleitung zum Programm sollte die üblichen »Zutaten« enthalten: Programmbeschreibung, Variablendefinition, Programmaufschlüsselung nach Zeilennummern, Flußdiagramm, bei Maschinensprache dokumentierter Quellcode, etc.
* Strukturierte Programmierung ist nicht erforderlich.

Zu den Spielprogrammen: Bei der Art des Programms sind wir nicht wählerisch, das heißt sie können aus dem gesamten Bereich des Spielespektrums von Schieß-, Kletter- und Hüpfspielen bis zu Denk- und Strategiespielen kommen. Wichtigste Voraussetzung: Alles muß mit dem oder den Joysticks zu steuern sein.

Zu den Lernprogrammen: Auch hier kann jede Art von Programmen, die irgend etwas mit Lernen zu tun haben, eingesandt werden. Die Rechte an den Programmen gehen an den Verlag Markt & Technik AG über. Schikken Sie Ihre Programme an:

Verlag Markt & Technik AG
**Stichwort: Btx-Wettbewerb**
Hans-Pinsel-Str. 2
8013 Haar bei München

## Einsendeschluß ist der 15. Juni 1985

Es gilt das Datum des Poststempels. Der Rechtsweg ist ausgeschlossen.




Die angegebenen Preise sind Ladenpreise.

## Inserentenverzeichnis




## Impressum


**Herausgeber:** Carl-Franz von Quadt, Otmar Weber

**Chefredakteur:** Michael M. Pauly (py)
**Stellv. Chefredakteur:** Michael Scharfenberger (sc)
**Redakteure:** aa = Albert Absmeier, leitender Redakteur, ah = Achim Hübner, ev = Volker Everts, gk = Georg Klinge, hm = Harald Meyer, rg = Christian Rogge
**Redaktionsassistenz:** Gerda Vogl (202)
**Fotografie:** Janos Feitser/Jens Jancke, Titelfoto: Jens Janckes

**Layout:** Leo Eder (Ltg.), Dagmar Berninger, Willi Gründl

**Auslandsrepräsentation:**
**Schweiz:** Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug, Tel. 042-223155/56, Telex: 862329 mut ch
**USA:** M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. (415) 424-0600; Telex 752351

**Manuskripteinsendungen:** Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

**Herstellung:** Klaus Buck (180)

**Anzeigenverkaufsleitung:** Ralph Peter Rauchfuss (126)

**Anzeigenleitung:** Christian Schlottau (282)

**Anzeigenverwaltung und Disposition:** Michaela Hörl (171)

**Anzeigenformate:** 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297x210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

**Anzeigenpreise:** Es gilt die Anzeigenpreisliste Nr. 2 vom 1. Januar 1985.
**Anzeigengrundpreise:** 1/1 Seite sw: DM 8500,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1400,-. Vierfarbzuschlag DM 3800,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/2-Seite
**Anzeigen im Computer-Markt:** Die ermäßigten Preise im Computer-Markt gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 6400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
**Gewerbliche Kleinanzeigen:** DM 11,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

**Vertriebsleitung, Werbung:** Hans Hörl (114)

**Vertrieb Handelsauflage:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

**Erscheinungsweise:** 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

**Bezugsmöglichkeiten:** Leser-Service: Telefon 089/4613-119. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

**Bezugspreise:** Das Einzelheft kostet DM 6,50. Der Abonnementspreis beträgt im Inland DM 78,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

**Druck:** E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall





**Verantwortlich:** Für redaktionellen Teil: Michael M. Pauly. Für Anzeigen: Christian Schlottau.

**Redaktions-Direktor:** Michael Pauly

**Vorstand:** Carl-Franz von Quadt, Otmar Weber

**Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:**
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon 089/4613-0, Telex 522052



**Telefon-Durchwahl im Verlag:**

**Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.**

## IEEE-Interface zum Selberbauen

Um beispielsweise Peripheriegeräte von den großen CBM-Rechnern (Floppy-Laufwerke, Drucker oder Meßgeräte) an den C 64 anschließen zu können, benötigen Sie ein IEEE-Interface. Die zur Zeit erhältlichen Geräte kosten meist über 200 Mark. Das hier beschriebene Interface ist zu einem Bauteilepreis von unter 50 Mark zu haben.

## Der Top-Assembler

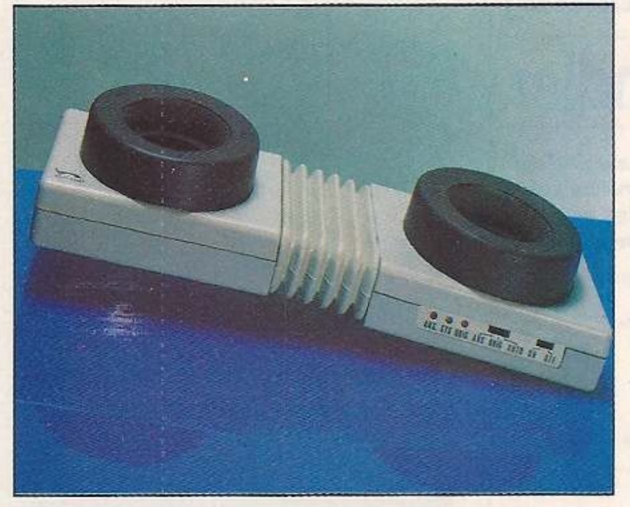
Zusammen mit dem SMON wird mit diesem Assembler dem Maschinenprogrammierer ein Programmpaket zur Verfügung gestellt, das leistungsfähiger ist als viele kommerziell angebotene Programme. Die hervorstechenste Eigenschaft dieses Assemblers ist die Möglichkeit, Macros anzulegen, in denen lokale Variablen benutzt werden können. Neben vielen Pseudo-Opcodes kann auch bedingt assembliert werden. Ein eingebautes Hilfstool enthält Befehle wie zum Beispiel Auto, Renumber und formatiertes Listen. Ein Listing des Monats par excellence.

## Software-Test Promal, Forth und Pilot

Drei ganz unterschiedliche Programmiersprachen für den C 64: Pil'ot wurde in erster Linie für Ausbildungszwecke entwickelt, Forth stammt aus der Steuerungstechnik, und bei Promal handelt es sich um eine völlig neue Sprache, die in der Struktur an Pascal angelehnt ist. Alle drei Programmiersprachen haben eines gemeinsam: Sie ermöglichen sehr effektives Programmieren — das bedeutet kurze, schnelle Programme, die wenig Speicherplatz benötigen, aber viel leisten. Wir haben diese drei Sprachen daher für Sie ausführlich getestet.

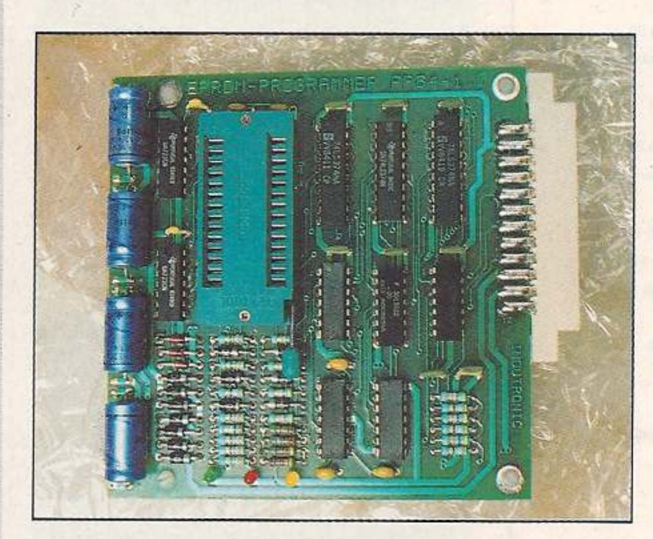
## Außerdem ...

— ein File-Compactor
— eine Centronics-Schnittstelle ohne Speicherplatzverbrauch
— ein Bericht, wie der 64er für die Auswertung von internationalen Sportveranstaltungen eingesetzt wird
— sowie wieder viele Tips & Tricks für C 16, VC 20 und C 64



## Modems und Akustikkoppler

Das Angebot der Modems und Akustikkoppler ist seit unserer letzten Marktübersicht um einiges größer geworden. Es reicht von einfachen Platinen bis zu preiswerten und leistungsstarken Geräten. Auf welche Leistungsmerkmale man achten sollte, zeigt unser Bericht.




## Was ist ein EPROM?

Möchten Sie das Betriebssystem des C 64 ändern, ist die Anschaffung eines EPROM-Programmierers unumgänglich. In einem Test stellen wir Ihnen neben einer ausführlichen Funktionsbeschreibung in einer Marktübersicht die zur Zeit erhältlichen Geräte für den C 64 vor.




## Die anderen Drucker

Was gibt Tintenstrahldruckern und Plottern den Ruf des Besonderen? In der nächsten Ausgabe sagen wir Ihnen, was an diesem Ruf »dran« ist. Das Prinzip der Billigsten, der Thermodrucker, wird ebenfalls erklärt. Eine Marktübersicht dieser Druckertypen soll Ihnen beim Kauf helfen.

## Test: PC 128 (2. Teil)

Im zweiten Teil unseres großen PC 128-Tests finden Sie alles über den neuen RGB/Composite-Farbmonitor 1902 und über das schnelle 1571-Floppy-Laufwerk, das dem PC 128 auch zu CP/M verhilft. Das ist das Stichwort zum zweiten großen Thema unseres Testberichts. Sie erfahren, was CP/M ist, wie es auf dem PC 128 funktioniert und natürlich auch, welche Software damit läuft.

## Terminalprogramm zum Abtippen

Viele gute Programme erreichten die Redaktion zum Wettbewerb Terminalprogramme. Ein Programm jedoch ließ die anderen weit hinter sich. Auch den Vergleich mit professionellen Terminalprogrammen braucht es nicht zu scheuen. Überzeugen Sie sich in der nächsten Ausgabe selbst.


64ER ONLINE


64ER ONLINE